ABBYY®

ABBYY Mobile Capture SDK

React Native Module

Table of Contents

Introduction

ABBYY Mobile Capture SDK React Native Module allows to use ABBYY Mobile Capture SDK features in apps based on the React Native framework. Image Capture feature is available as a complete scenario, including user interface implementation. Core API methods for low-level single image processing, such as data extraction, text recognition, images export, etc., can be simply integrated into your projects.

This module requires the ABBYY Mobile Capture assets, which differ for Android and iOS, and a license file. You can request ABBYY Mobile Capture trial version on the ABBYY website.

This manual describes JavaScript API. More information, including the library usage details, is available in the *ABBYY Mobile Capture SDK Developer's Guide* found in the library packages.

Getting Started

To start developing with the ABBYY Mobile Capture SDK React Native Module, you need to add the module and ABBYY Mobile Capture to your project. Find the steps below.

Note: *If you are new to React Native, set up development environment first. You will need* ***yarn****,* ***npx*** *and* ***cocoapods*** *(for iOS) utilities. See [React Native documentation] for details.*

1. Choose one of the following ways to add the module to your project:

- From npm:

```
yarn add react-native-abbyy-mobile-capture
```

- From sources:

```
yarn add file:"Path to react-native-abbyy-mobile-capture sources"
```

2. Request ABBYY Mobile Capture trial version on the ABBYY website and add it to your project:

- Copy iOS **assets** directory to the project into **./assets** folder
- Copy your license into **./assets** with **MobileCapture.License** name
- Copy Android **libs** directory to the project into **./android/libs** folder
- Copy iOS **libs** directory to the project into **./ios/libs** folder

3. Proceed setup depending on the platform:

- **Android setup**

  - Add **gradle** file to the project. Usually it is located at **./android/build.gradle**:

```
allprojects {
   repositories {
      flatDir {
      dirs "$rootDir/libs"
      }
   }
}
subprojects {
   afterEvaluate {
      dependencies {
         // Change abbyy-mi-sdk-2.0 to abbyy-rtr-sdk-1.0 if you need to
use recognition API (MI+Extended or MI+ExtendedDC)
         implementation(name: 'abbyy-mi-sdk-2.0', ext: 'aar')
         implementation(name: 'abbyy-ui-1.0', ext: 'aar')
      }
   }
}
```

  - Add to application **gradle** file. Usually it is located at **./android/app/build.gradle**:

```
android {
         defaultConfig {
            ndk {
                  // Filter out mips
                  abiFilters 'armeabi-v7a', 'arm64-v8a', 'x86', 'x86_64'
              }
         }

         sourceSets {
            main {
               assets.srcDirs += [ "$rootDir/../assets" ]
            }
         }

         packagingOptions {
            pickFirst 'lib/x86/libc++_shared.so'
            pickFirst 'lib/armeabi-v7a/libc++_shared.so'
            pickFirst 'lib/arm64-v8a/libc++_shared.so'
            pickFirst 'lib/x86_64/libc++_shared.so'
         }
      }
```

  - Set minimum SDK version to '21'.
  - Add **adb** utility to PATH environment variable. **adb** is located in **Android sdk directory/platform-tools** directory.

- **iOS setup**
  - Run **pod install** from iOS directory.
  - Select your project in the **Target** group and open the **Build Settings** tab. In the **Search Paths** section add to the **Framework Search Paths** the following path:

```
${PROJECT_DIR}/libs
```

  - On the **Build Options** tab set **Enable Bitcode** option to **No**.
  - Open the **Build Phases** tab. In the **Link Binary With Libraries** section, click "+" and add the **AbbyyRtrSDK.framework** and **AbbyyUI.framework**.
  - In **Build Phases**, add a new **Run Script** phase. Run the **copy_frameworks.sh** script that removes the frameworks for the non-active CPU architectures (the complete list depends on the project settings), and sign the resulting framework. This is a required step before uploading your application to App Store.

```
/bin/sh "${SRCROOT}/libs/copy_frameworks.sh"
```

  - To add the resource files and set up the copying rules, in **Build Phases** add one more **Run Script** phase. Run the **copy_assets.py** script to automatically copy all resource files to corresponding destinations and add necessary dictionaries.

```
python "${SRCROOT}/../assets/copy_assets.py"
```

- In **Build Phases**, go to the **Copy Bundle Resources** section and add the **MobileCapture.License** file.
- Add the NSCameraUsageDescription and NSPhotoLibraryUsageDescription keys into the **info.plist** file for requesting access to the device's camera and to the user's photo library.

How to run samples

1. Request ABBYY Mobile Capture trial version on the ABBYY website and add it to the sample:

- Copy iOS **assets** directory to the project into **./assets** folder
- Copy your license file into **./assets** with **MobileCapture.License** name
- Copy Android **libs** directory to the project into **./android/libs** folder
- Copy iOS **libs** directory to the project into **./ios/libs** folder

2. Run **yarn install** from the sample root

3. Run **pod install** from the sample/ios directory

4. To run application do the following:

Android	**iOS**
• From console: ○ Execute **npx react-native run-android** from the sample root • From Android Studio: ○ Open **sample/android** project ○ Click **Run app** button	• From console: ○ Execute **npx react-native run-ios** from the sample root It runs sample in the simulator • From XCode ○ Open **sample/ios** project ○ Click **Run** button

AbbyyMobileCapture module

ABBYY Mobile Capture SDK module.

Methods

Name	Description
User interface	
startImageCapture	Opens a screen for the Image Capture scenario.
Core API	
recognizeText	Starts a text capture scenario for a single image.
extractData	Starts a data capture scenario for a single image.
detectDocumentBoundary	Detects a quadrangle representing document boundary on an image.
cropImage	Crops image according to the document boundary and size.
rotateImage	Rotates image by specified angle.
assessQualityForOcr	Estimates if an image quality is suitable for OCR.
exportImage	Exports an image to JPG or PNG format.
exportImagesToPdf	Exports images to PDF format.

startImageCapture method

Opens a screen for the Image Capture scenario.

Note: *You can manage and localize UI string sources. See User interface string sources section for details.*

```
async function startImageCapture(settings)
```

Parameters

settings

Object specifying image capture parameters (see Options).

Return values

The method returns result depending on how the scenario finished (see Result).

Options

The table below describes parameters that you can pass as the *settings* argument to change image capture settings. All parameters are optional. Omitting a parameter means that a default setting will be used.

Parameter	Value type	Description
licenseFileName	string	The name of the license file. This file must be located in the **/assets/** directory in your project. Default: "MobileCapture.License".
cameraResolution	string	The resolution of the images captured from the camera preview. This parameter can be set to the following values: • "HD" • "FullHD" • "UHD_4K" Default: "FullHD".
isFlashlightButtonVisible	boolean	Show (true) or hide (false) the flashlight button in the image capture dialog. Default: true (flashlight visible).
isCaptureButtonVisible	boolean	Show (true) or hide (false) the button for taking photo manually. Default: true (button is visible).
isGalleryButtonVisible	boolean	Show (true) or hide (false) the button for choosing an image from photo gallery. Default: true (button is visible).

Parameter	Value type	Description
orientation	string	The type of image orientation. This parameter can be set to the following values: • "Portrait" • "Landscape" • "Default" (the orientation will be detected automatically) Default: "Default".
isShowPreviewEnabled	boolean	Show a preview of an image after capture (true) or show a camera view (false) Default: false (camera view is shown).
requiredPageCount	int	Total number of pages to be captured. Set the page-limitation mode of the image capture as following: • 0 to allow unlimited image capture. The set of the result images can be saved or edited at any time; • a positive value to set the exact number of images that should be captured. Images saving is enabled only when this number of images are been captured. Default: 0.
destination	string	Captured image will be saved to corresponding file ("File") or returned as encode base64 image string ("Base64"). ❗ ***Note:*** *Image can be saved to Base64 only for requiredPageCount=1. Otherwise an exception will be thrown.* Default: "File".
exportType	string	Captured image will be saved to this format. This parameter can be set to the following values: • "Jpg" • "Png" • "Pdf" (all images will be saved to one PDF file) Default: "Jpg".
compressionLevel	string	The uniform image compression scale for lossy formats. This parameter can be set to the following values: • "Low" • "Normal" • "High"

Parameter	Value type	Description
		- "ExtraHigh" Default: "Low".
defaultImageSettings	object	Custom image capture settings. This object has the following parameters: - **aspectRatioMin** (float): Lower limit of the document's aspect ratio. This property is used in pair with the **aspectRatioMax**, defining an interval of acceptable aspect ratio values of the document to be captured. Setting aspect ratio will help to improve boundary detection accuracy. If only **aspectRatioMax** is set, **aspectRatioMin** will be set to 1. Default: 0 (aspect ratio is not set). - **aspectRatioMax** (float): Upper limit of document's aspect ratio. This property is used in pair with the **aspectRatioMin**, defining an interval of acceptable aspect ratio values of the document to be captured. Setting aspect ratio will help to improve boundary detection accuracy. If only **aspectRatioMin** is set, **aspectRatioMax** will be set to infinity. Default: 0 (aspect ratio is not set). - **imageFromGalleryMaxSize** (int): Maximum available size of an image, loaded from the gallery. The size is defined as the length of the largest side of an image (in pixels). Default: 4096. - **minimumDocumentToViewRatio** (float): the minimum document area relative to the whole frame area, required for capture. If the document area is less than this ratio, the image will not be captured. The value can be from 0 to 1. Default: 0.15. - **documentSize** (string): expected size of the original document in millimeters. This parameter can be set to the following constant values: ○ "A4" ○ "BusinessCard" ○ "Letter" ○ "Any" ○ or defined as a string with 2 integers separated with an 'x' symbol: "100x100". All parameters are optional.

Result

This section describes the object that represents image capture results. Returned parameters depend on the scenario.

Parameter	Value type	Description
images	object array	Captured images. This parameter is returned only if the **exportType** is set to "Jpg" or "Png". Each object in the array represents a single image with corresponding information. The image objects have the following parameters: • **resultInfo** (object): this object has the following parameters: ○ **exportType** (string): the format of the captured image. ○ **imageSize** (object): full size of the preview frame, an object with int parameters **width** and **height**. ○ **uriPrefix** (string): add this string to the beginning of the **filePath** or **base64** parameter to get a URI. This parameter can be 'file://' if the destination is 'File' and 'data:image/jpeg;base64,' or 'data:image/png;base64,' if the destination is 'Base64'. • **filePath** (string): full path to the exported file. This parameter is present if the destination is 'File'. • **base64** (string): the original image returned as base64 image string. This parameter is present if the **destination** parameter was set to "Base64".
pdfInfo	object	This parameter is returned if the **exportType** is set to "Pdf". The value is an object which has the following parameters: • **filePath** (string): full path to the exported PDF file. • **pagesCount** (integer): number of pages in the exported PDF file. Normally fits the number of captured images.
resultInfo	object	Additional information. This object has the following parameters: • **userAction** (string): the user's action which stopped the module, if any. Can be "Canceled" if the user canceled processing. If the module has stopped automatically, the **userAction** parameter is not present in **resultInfo**. • **uriPrefix** (string): add this string to the beginning of the **filePath** or **base64** parameter to get a URI. This parameter can be 'file://' if the destination is 'File' and 'data:image/jpeg;base64,' or 'data:image/png;base64,' if the destination is 'Base64'.

Below is an example of a result JSON when image capture succeeds.

```
{
    "images": [
        {
            "resultInfo": {
                imageSize: {
                  width: 658,
                  height: 1187
                }
               "exportType": "Jpg"
            },
            "filePath":
"/data/user/0/com.abbyy.rtr.reactnative.sample/files/pages/page_334fd281-
f472-4756-a4a0-d1f8d1857a0c.jpg"
        }
    ]
}
```

User interface string sources

The user interface messages can be adjusted depending on the application language and specific. The ways differ for different platforms. Choose the one appropriate for your application:

Android

Add the following attributes to the **strings.xml** file and edit as needed.

```
<string name="uic_camera_need_permission_tip_title">Permissions
required</string>
<string name="uic_camera_need_permission_tip_message">Please allow the
app to access the camera on this device</string>

<string
name="uic_camera_dialog_permission_rationale_positive_button">OK</string
>
<string
name="uic_camera_dialog_permission_rationale_negative_button">Cancel</st
ring>
<string name="uic_camera_dialog_permission_rationale_title">Permissions
denied</string>
<string name="uic_camera_dialog_permission_rationale_message">This app
needs access to the camera on this device</string>

<string
name="uic_camera_dialog_permission_settings_positive_button">Open
settings</string>
<string
name="uic_camera_dialog_permission_settings_negative_button">Cancel</str
ing>
<string name="uic_camera_dialog_permission_settings_title">Permissions
denied</string>
<string name="uic_camera_dialog_permission_settings_message">Please
allow the app to access the camera.\nYou can turn on camera permissions
in the system settings.</string>

<string name="uic_ics_looking_for_document_tip">Looking for
document</string>
<string name="uic_ics_move_closer_tip">Closer</string>
<string name="uic_ics_dont_move_tip">Keep still</string>
```

```
<string name="uic_mpics_camera_number_of_captured_pages_fixed_count">%1$d of %2$d</string>
<plurals name="uic_mpics_camera_number_of_captured_pages">
<item quantity="one">%d Page</item>
<item quantity="other">%d Pages</item>
</plurals>

<string name="uic_mpics_editor_add_page_button">Add Page</string>
<string name="uic_mpics_editor_next_document_button">Next Document</string>
<string name="uic_mpics_editor_done_button">Done</string>
<string name="uic_mpics_editor_current_document_index">%1$d of %2$d</string>

<string name="uic_mpics_crop_edit">Edit Crop</string>
<string name="uic_mpics_crop_auto">Auto crop</string>
<string name="uic_mpics_crop_full">Expand crop area to the full screen</string>
<string name="uic_mpics_crop_cancel">Cancel</string>
<string name="uic_mpics_crop_accept">Accept</string>

<string name="uic_mpics_editor_delete_all_warning_title">Delete all</string>
<string name="uic_mpics_editor_delete_all_warning_message">Are you sure?</string>
<string name="uic_mpics_editor_delete_all_warning_positive_button">Delete</string>
<string name="uic_mpics_editor_delete_all_warning_negative_button">Cancel</string>

<string name="uic_mpics_editor_delete_page_warning_title">Delete page</string>
<string name="uic_mpics_editor_delete_page_warning_message">Are you sure?</string>
<string name="uic_mpics_editor_delete_page_warning_positive_button">Delete</string>
<string name="uic_mpics_editor_delete_page_warning_negative_button">Cancel</string>

<string name="uic_mpics_editor_error_title">Error</string>
<string name="uic_mpics_editor_error_button">OK</string>
<string name="uic_mpics_editor_page_deleted">Page deleted</string>
<string name="uic_mpics_editor_add_more_pages">Add more pages</string>

<string name="uic_mpics_editor_retake">Retake</string>
<string name="uic_mpics_editor_delete_page">Delete page</string>
<string name="uic_mpics_editor_delete_all">Delete all</string>

<string name="uic_mpics_preview_title">Preview</string>

<string name="uic_rn_ic_saving_pages_warning_message">Saving pages...</string>
```

```
<string name="uic_rn_ic_delete_on_cancel_warning_title">Discard
pages</string>
<string name="uic_rn_ic_delete_on_cancel_warning_message">You will lose
captured pages</string>
<string
name="uic_rn_ic_delete_on_cancel_warning_negative_button">Cancel</string
>
<string
name="uic_rn_ic_delete_on_cancel_warning_positive_button">Discard</strin
g>
```

iOS

Run the **sample-ui-imagecapture/ios/sample-ui-imagecapture.xcworkspace** file and find user interface string sources in the **AbbyyUI.strings** and **MobileCapture.strings** files located in the **sample-ui-imagecapture** folder.

Core API methods

This section contains description of the low-level methods for single image processing.

recognizeText method

Starts text capture and recognition for a single image. Captured image will be processed according to the passed settings and recognized text will be returned with some technical information such as warnings, if any, and text orientation on the image.

```
export async function recognizeText(settings)
```

Parameters

settings

Object specifying parameters of the text capture scenario for a single image (see Options).

Return values

The method returns result depending on how the scenario finished (see Result).

Options

The table below describes parameters that you can pass as the *settings* argument to change text capture settings. Omitting a parameter means that a default setting will be used.

Parameter	Value type	Description
licenseFileName	string	The name of the license file. This file must be located in the **/assets/** directory in your project. Default: "MobileCapture.License".
imageUri	string	**imageUri** (string): Image source for the operation passed as URI. This parameter can be set to the following values: • "Base64": you can pass an encode base64 image string, i.e. "*data:image/jpeg;base64,<base64 encoded image data>*" • "File": image file address, i.e. "*file:///data/user/0/com.abbyy.rtr.reactnative.sample.coreapi/files/pages/page_848b121d-5a7a-4ead-94e6-dd91bc6bfead.jpg*". '*content://*' schema is also supported for Android. Required parameter.
areaOfInterest	object	A rectangle specifying the area of interest in the coordinates, set by the following parameters: • **top** (int): upper left point ordinate • **bottom** (int): bottom left point ordinate • **left** (int): upper left point abscissa • **right** (int): upper right point abscissa I.e., {top: 100, bottom: 1000, left: 100, right: 1000} ***Note:*** *All parameters should be defined to set the* ***areaOfInterest*** *parameter. If some of them are defined and some are not, an error occurs.* Default: a rectangle, containing the whole image.
isTextOrientationDetectionEnabled	boolean	Enables or disables detection of the image orientation while preprocessing. If the property is set to **true**, the image top is detected and correct orientation can be used for image rotation. You can set this property to **false** for speeding the process up. ***Note:*** *Disable the image detection only if you can be sure that the captured image has correct orientation. Otherwise the text on image will not be detected and recognized.*

Parameter	Value type	Description
		The default value of this property is **true** (enabled).
recognitionLanguages	string[]	List of languages, supported for the text recognition, i.e. ['English', 'Russian']. See the full list of supported languages here. Default: ['English'].

Result

This section describes the object that represents text capture results. Returned parameters depend on the captured and passed image.

Parameter	Value type	Description
orientation	int	An angle on which the image was rotated to get normal orientation. Possible values are: 0, 90, 180, 270.
warnings	string[]	Warnings that occurred during processing, if any.
text	string	The whole recognized text as a single string. Text lines are divided with "\n". Text blocks are divided with "\n\n".
textBlocks	object[]	An array of text blocks. Each text block is represented as an array of **textLines** parameter.
textBlocks.textLines	object[]	An array of text lines. Each text line has the following parameters: • **text** (string): recognized text. • **quadrangle** (object): quadrangle containing text, represented by an array of its four vertices' coordinates: { **x**: int, **y**: int }. The vertices are indexed clockwise starting from the bottom left. • **rect** (object): bounding rectangle for quadrangle. The rectangle is set by the following parameters: • **top** (int): upper left point ordinate • **bottom** (int): bottom left point ordinate • **left** (int): upper left point abscissa • **right** (int): upper right point abscissa • **charInfo** (object[]): an array, which elements contain information concerning concrete recognized symbol.

Parameter	Value type	Description
textBlocks.textLines.charInfo	object[]	An array, which elements contain information concerning concrete recognized symbol. This object contains parameters, describing the symbol's place at an image, and its recognition confidence. - **quadrangle** (object): quadrangle containing text, represented by an array of its four vertices' coordinates: { **x**: int, **y**: int } - **rect** (object): bounding rectangle for quadrangle. The rectangle is set by the following parameters: - **top** (int): upper left point ordinate - **bottom** (int): bottom left point ordinate - **left** (int): upper left point abscissa - **right** (int): upper right point abscissa - **isUncertain** (boolean): returned only in case the symbol is uncertain.

Example of a result JSON.

```
{"textBlocks": [{
  "textLines": [
   {
     "text": "",
     "quadrangle": [
       {
        "x": 82,
        "y": 567
      },
       {
        "x": 82,
        "y": 567
      },
       {
        "x": 82,
        "y": 567
      },
       {
        "x": 82,
        "y": 567
      }
     ],
     "rect": {
       "top": 515,
       "bottom": 568,
       "left": 82,
       "right": 528
     },
     "charInfo": [
       {
        "quadrangle": [
         {
           "x": 82,
           "y": 567
         },
         {
           "x": 82,
           "y": 567
         },
         {
           "x": 82,
           "y": 567
         },
         {
           "x": 82,
           "y": 567
         }
        ],
        "rect": {
         "top": 515,
         "bottom": 568,
         "left": 82,
         "right": 528
        },

        "isUncertain": true,
        "isItalic": true
      }
     ],
     "text": "",
     "orientation": 0
    }]
  }]
}
```

extractData method

Starts data capture for a single image. Data fields will be detected and recognized on the captured image according to the passed settings. One of the settings is a date capture profile, representing a type of a document to be recognized, i.e. a business card. The profile defines a data schema, describing document fields. The profile schema is applied during the capture process.

Result will be returned with some technical information such as warnings, if any, and text orientation on the image.

❗ ***Note:*** *The functionality is currently supported for business cards recognition only.*

```
export async function extractData(settings)
```

Parameters

settings

Object specifying parameters of the data capture scenario for a single image (see Options).

Return values

The method returns result depending on how the scenario finished (see Result).

Options

The table below describes parameters that you can pass as the *settings* argument to change data capture settings. Omitting a parameter means that a default setting will be used.

Parameter	Value type	Description
licenseFileName	string	The name of the license file. This file must be located in the **/assets/** directory in your project. Default: "MobileCapture.License".
imageUri	string	**imageUri** (string): Image source for the operation passed as URI. This parameter can be set to the following values: • "Base64": you can pass an encode base64 image string, i.e. "*data:image/jpeg;base64,<base64 encoded image data>*" • "File": image file address, i.e. "*file:///data/user/0/com.abbyy.rtr.reactnative.sample.coreapi/files/pages/page_848b121d-5a7a-4ead-94e6-dd91bc6bfead.jpg*". '*content://*' schema is also supported for Android. Required parameter.
profile	string	The profile name. I.e. 'BusinessCards'. ❗ ***Note:*** *The functionality is currently supported for business cards recognition only.*
isTextOrientationDetectionEnabled	boolean	Enables or disables detection of the image orientation while preprocessing. If the property is set to **true**, the image top is detected and correct orientation can be used for image rotation. You can set this property to **false** for speeding the process up. ❗ ***Note:*** *Disable the image detection only if you*

Parameter	Value type	Description
		can be sure that the captured image has correct orientation. Otherwise the text on image will not be detected and recognized. The default value of this property is **true** (enabled).
recognitionLanguages	string[]	List of languages, supported for the text recognition, i.e. ['English', 'Russian']. See the full list of supported languages here. Default: ['English'].

Result

This section describes the object that represents data capture results. Returned parameters depend on the captured and passed image.

Parameter	Value type	Description
orientation	int	An angle on which the image was rotated to get normal orientation. Possible values are: 0, 90, 180, 270.
warnings	string[]	Warnings that occurred during processing, if any.
dataFields	object[]	List of objects representing captured and recognized data fields. Each data field has the following parameters: • **id** (int): internal identifier of the data field. • **name** (string): human readable name of the data field. • **text** (string): recognized text of the data field. • **quadrangle** (object): quadrangle containing text, represented by an array of its four vertices' coordinates: { **x**: int, **y**: int }. The vertices are indexed clockwise starting from the bottom left. • **rect** (object): bounding rectangle for quadrangle. The rectangle is set by the following parameters: • **top** (int): upper left point ordinate • **bottom** (int): bottom left point ordinate • **left** (int): upper left point abscissa • **right** (int): upper right point abscissa • **components** (object[]): field components. If the field has only one component, this array contains one element. This property may be 0, if the field is not compound. In

Parameter	Value type	Description
		this case result does not contain **components**. Each component can have the same parameters as **dataField** including **components** array. • **charInfo** (object[]): an array, which elements contain information concerning concrete recognized symbol.
dataFields.charInfo	object[]	An array, which elements contain information concerning concrete recognized symbol. This object contains parameters, describing the symbol's place at an image, and its recognition confidence. • **quadrangle** (object): quadrangle containing text, represented by an array of its four vertices' coordinates: { **x**: int, **y**: int } • **rect** (object): bounding rectangle for quadrangle. The rectangle is set by the following parameters: • **top** (int): upper left point ordinate • **bottom** (int): bottom left point ordinate • **left** (int): upper left point abscissa • **right** (int): upper right point abscissa • **isUncertain** (boolean): returned only in case the symbol is uncertain.

Example of a result JSON.

```
{"dataFields": [
  {
    "id": "Mobile",
   "name": "Mobile",
   "text": "",
    "quadrangle": [
     {
         "x": 82,
       "y": 567
     },
     {
       "x": 82,
      "y": 567
     },
     {
       "x": 82,
       "y": 567
     },
     {
       "x": 82,
       "y": 567
     }
    ],
    "rect": {
     "top": 515,
     "bottom": 568,
     "left": 82,
     "right": 528
    },
    "charInfo": [
     {
       "quadrangle": [
        {
          "x": 82,
          "y": 567
        },
        {
          "x": 82,
          "y": 567
        },
        {
          "x": 82,
          "y": 567
        },
        {
          "x": 82,
          "y": 567
        }
       ],
       "rect": {
        "top": 515,
        "bottom": 568,
        "left": 82,
        "right": 528
       },
       "isUncertain": true
     }
    ],
   "components": [
     {
      "id": "PhoneCode",
      "name": "PhoneCode",
      "text": "",
      "quadrangle": [
        {
         "x": 82,
         "y": 567
        },
        {
         "x": 82,
         "y": 567
        },
        {
         "x": 82,
         "y": 567
        },
        {
         "x": 82,
         "y": 567
        }
      ],
      "rect": {
        "top": 515,
        "bottom": 568,
        "left": 82,
        "right": 528
      },
      "charInfo": [
        {
         "quadrangle": [
           {
            "x": 82,
            "y": 567
           },
           {
            "x": 82,
            "y": 567
           },
           {
            "x": 82,
            "y": 567
           },
           {
            "x": 82,
            "y": 567
           }
         ],
         "rect": {
           "top": 515,
           "bottom": 568,
           "left": 82,
           "right": 528
         },
         "isUncertain": true
        }
      ]
     }
   ]
  }]
}
```

detectDocumentBoundary method

Detects a quadrangle representing document boundary on an image.

```
export async function detectDocumentBoundary(settings)
```

Parameters

settings

Object specifying parameters of the document boundary detection (see Options).

Return values

The method returns result depending on how the document boundary detection finished (see Result).

Options

The table below describes parameters that you can pass as the *settings* argument to change document boundary detection settings. Omitting a parameter means that a default setting will be used.

Parameter	Value type	Description
licenseFileName	string	The name of the license file. This file must be located in the **/assets/** directory in your project. Default: "MobileCapture.License".
imageUri	string	**imageUri** (string): Image source for the operation passed as URI. This parameter can be set to the following values: • "Base64": you can pass an encode base64 image string, i.e. "*data:image/jpeg;base64,<base64 encoded image data>*" • "File": image file address, i.e. "*file:///data/user/0/com.abbyy.rtr.reactnative.sample.coreapi/files/pages/page_848b121d-5a7a-4ead-94e6-dd91bc6bfead.jpg*". '*content://*' schema is also supported for Android. Required parameter.
areaOfInterest	object	A rectangle specifying the area of interest in the coordinates, set by the following parameters: • **top** (int): upper left point ordinate • **bottom** (int): bottom left point ordinate • **left** (int): upper left point abscissa • **right** (int): upper right point abscissa I.e., {top: 100, bottom: 1000, left: 100, right: 1000} ❗ ***Note:*** *All parameters should be defined to set the* ***areaOfInterest*** *parameter. If some of them are defined and some are not, an error occurs.* Default: a rectangle, containing the whole image.
detectionMode	string	Document boundary detection mode. The mode influences the crop speed and accuracy. This parameter can be set to the following values:

Parameter	Value type	Description
		○ 'Fast' - this mode signifies processing speed ○ 'Default' - balanced mode, that combines optimal processing speed and high quality. Default: 'Default'.
documentSize	object	Document size represented by its width and height in millimeters. The values of the width and height are set to the **documentSize** parameters: • **width** (float) • **height** (float) I.e., {x: 210; y: 297} for A4 document size. To leave the document size undefined, set both width and height to 0. In this case document of any size will be detected.

Result

This section describes the object that represents image capture results. Returned parameters depend on the scenario.

Parameter	Value type	Description
documentSize	object	Detected document size represented by its width and height in millimeters. The values of the width and height are stored in the **documentSize** parameters: • **width** (int) • **height** (int)
documentBoundary	object	Quadrangle containing the whole document. It is represented by an array of its four vertices' coordinates: { **x**: int, **y**: int }. The vertices are indexed clockwise starting from the bottom left. Default: coordinates of a quadrangle, containing the whole image. When document boundary is not detected, this parameter is not returned.

Example of a result JSON.

```
{
  "documentSize": {
    "width": 0,
   "height": 0
  },
  "documentBoundary": [
  {
   "x": 82,
   "y": 567
  },
  {
   "x": 82,
   "y": 567
  },
  {
   "x": 82,
   "y": 567
  },
  {
   "x": 82,
   "y": 567
  }
  ]
}
```

cropImage method

Crops image according to the document boundary and size. Applying this operation removes empty fields around a detected document the and corrects perspective distortion if needed.

```
export async function cropImage(settings)
```

Parameters

settings

Object specifying parameters of the image crop operation (see Options).

Return values

The method returns result depending on how the image crop finished (see Result).

Options

The table below describes parameters that you can pass as the *settings* argument to change image crop settings. Omitting a parameter means that a default setting will be used.

Parameter	Value type	Description
licenseFileName	string	The name of the license file. This file must be located in the **/assets/** directory in your project. Default: "MobileCapture.License".
imageUri	string	**imageUri** (string): Image source for the operation passed as URI. This parameter can be set to the following values: • "Base64": you can pass an encode base64 image string, i.e. "*data:image/jpeg;base64,<base64 encoded image data>*" • "File": image file address, i.e. "*file:///data/user/0/com.abbyy.rtr.reactnative.sample.coreapi/files/pages/page_848b121d-5a7a-4ead-94e6-dd91bc6bfead.jpg*". '*content://*' schema is also supported for Android. Required parameter.
result	object	Result image description, represented by the following parameters: • **compressionLevel** (string): the uniform image compression scale for lossy formats. Valid for export to JPG only. This parameter can be set to the following values: • 'Low' • 'Normal' • 'High' • 'ExtraHigh' Default: 'Low'. • **exportType** (string): result image will be saved to this format. This parameter can be set to the following values: ○ 'Jpg' ○ 'Png' Default: "Jpg". • **destination** (string): result image will be saved to corresponding file ("File") or returned as encode base64 image string ("Base64"). Default: "Base64". • **filePath** (string): full path to the exported file. This parameter is used only if the **destination** is 'File'. Otherwise the value will be ignored. In case the **destination** is 'File' and this parameter is not set, the path will be generated automatically.
documentSize	string	Document size represented by its width and height in

Parameter	Value type	Description
		millimeters. The values of the width and height are set to the **documentSize** parameters: • **width** (float) • **height** (float) I.e., {x: 210; y: 297} for A4 document size. To leave the document size undefined, set both width and height to 0. In this case document of any size will be detected.
documentBoundary	object	Quadrangle containing the whole document. It is represented by an array of its four vertices' coordinates: { **x**: int, **y**: int }. The vertices are indexed clockwise starting from the bottom left. Default: coordinates of a quadrangle, containing the whole image.

Result

This section describes the object that represents image capture results. Returned parameters depend on the scenario.

Parameter	Value type	Description
imageUri	string	Cropped image returned as URI. This image corresponds to the description in the passed **result** parameter.
resolution	object	The image resolution as calculated from image size and physical page size.
imageSize	object	Cropped image size represented by its width and height in pixels. The values of the width and height are stored in the following parameters: • **width** (int) • **height** (int)

Example of a result JSON.

```
{
  "imageUri": "file://data/user/0/com.abbyy.rtr.cordova.sample/files/
{guid}.jpg",
  "resolution": {
   "x": 0,
   "y": 0
  },
  "imageSize": {
    "width": 1966,
   "height": 3474
  }
}
```

rotateImage method

Rotates image by specified angle.

```
export async function rotateImage(settings)
```

Parameters

settings

Object specifying parameters of the image rotation (see Options).

Return values

The method returns result depending on how the scenario finished (see Result).

Options

The table below describes parameters that you can pass as the *settings* argument to change image rotation settings. Omitting a parameter means that a default setting will be used.

Parameter	Value type	Description
licenseFileName	string	The name of the license file. This file must be located in the **/assets/** directory in your project. Default: "MobileCapture.License".
imageUri	string	**imageUri** (string): Image source for the operation passed as URI. This parameter can be set to the following values: • "Base64": you can pass an encode base64 image string, i.e. "*data:image/jpeg;base64,<base64 encoded image data>*" • "File": image file address, i.e. "*file:///data/user/0/com.abbyy.rtr.reactnative.sample.coreapi/files/pages/page_848b121d-5a7a-4ead-94e6-*

Parameter	Value type	Description
		dd91bc6bfead.jpg". '*content://*' schema is also supported for Android. Required parameter.
angle	int	The angle in degrees. Available values of the angle: 0, 90, 180, 270.
result	object	Result image description, represented by the following parameters: • **compressionLevel** (string): the uniform image compression scale for lossy formats. Valid for export to JPG only. This parameter can be set to the following values: • 'Low' • 'Normal' • 'High' • 'ExtraHigh' Default: 'Low'. • **exportType** (string): result image will be saved to this format. This parameter can be set to the following values: ○ 'Jpg' ○ 'Png' Default: "Jpg". • **destination** (string): result image will be saved to corresponding file ("File") or returned as encode base64 image string ("Base64"). Default: "File". • **filePath** (string): full path to the exported file. This parameter is present if the destination is 'File'.

Result

This section describes the object that represents image capture results. Returned parameters depend on the scenario.

Parameter	Value type	Description
imageUri	string	Rotatedimage returned as URI. This image corresponds to the description in the passed **result** parameter.
size	object	Rotated image size represented by its width and height in pixels. The values of the width and height are stored in the

Parameter	Value type	Description
		following parameters: • **width** (int) • **height** (int)

Example of a result JSON.

```
{
  "imageUri": "file://data/user/0/com.abbyy.rtr.cordova.sample/files/
{guid}.jpg",
  "imageSize": {
    "width": 1966,
   "height": 3474
  }
}
```

assessQualityForOcr method

Estimates if an image quality is suitable for OCR. The whole image is represented as a set of rectangles. A type of rectangle can be either text or unknown. The quality assessment of the image for OCR is calculated based on the rectangles collection.

Note: *This is a technology preview feature. The functionality will be improved and completed in future versions.*

```
export async function assessQualityForOcr(settings)
```

Parameters

settings

Object specifying parameters of the image quality assessment (see Options).

Return values

The method returns result depending on how the scenario finished (see Result).

Options

The table below describes parameters that you can pass as the *settings* argument to change image quality assessment settings. Omitting a parameter means that a default setting will be used.

Parameter	Value type	Description
licenseFileName	string	The name of the license file. This file must be located in the **/assets/** directory in your project. Default: "MobileCapture.License".
imageUri	string	Image source for the operation passed as URI. This parameter can be set to the following values: • "Base64": you can pass an encode base64 image string, i.e. "*data:image/jpeg;base64,<base64 encoded image data>*" • "File": image file address, i.e. "*file:///data/user/0/com.abbyy.rtr.reactnative.sample.coreapi/files/pages/page_848b121d-5a7a-4ead-94e6-dd91bc6bfead.jpg*". '*content://*' schema is also supported for Android. Required parameter.
documentBoundary	object	Quadrangle containing the whole document. It is represented by an array of its four vertices' coordinates: { **x**: int, **y**: int }. The vertices are indexed clockwise starting from the bottom left. Default: coordinates of a quadrangle, containing the whole image.

Result

This section describes the object that represents image capture results. Returned parameters depend on the scenario.

Parameter	Value type	Description
qualityAssessmentForOcrBlocks	object[]	Collection of rectangles with corresponding quality assessment for each of them. Each rectangle can have the following parameters: • **type** (string): rectangle type for quality for OCR estimating. Can be 'Text' or 'Unknown'. • **quality** (int): for blocks with 'Text' **type** this parameter stores a value from 0 to 100 that indicates suitability of the text for OCR. • **rect** (object): bounding rectangle for quadrangle. The rectangle is set by the following parameters: • **top** (int): upper left point ordinate

Parameter	Value type	Description
		- **bottom** (int): bottom left point ordinate - **left** (int): upper left point abscissa - **right** (int): upper right point abscissa

Example of a result JSON.

```
{
  "qualityAssessmentForOcrBlocks": [
    {
     "type": "Text",
     "quality": 2,
     "rect": {
       "top": 1,
      "bottom": 65,
      "left": 65,
      "right": 129
     }
   }
  ]
}
```

exportImage method

Exports an image to JPG or PNG format. Please note, that this method exports a single image only.

```
export async function exportImage(settings)
```

Parameters

settings

Object specifying parameters of the image export (see Options).

Return values

The method returns result depending on how the scenario finished (see Result).

Options

The table below describes parameters that you can pass as the *settings* argument to change image export settings. Omitting a parameter means that a default setting will be used.

Parameter	Value type	Description
licenseFileName	string	The name of the license file. This file must be located in the **/assets/** directory in your project. Default: "MobileCapture.License".
imageUri	string	**imageUri** (string): Image source for the operation passed as URI. This parameter can be set to the following values: • "Base64": you can pass an encode base64 image string, i.e. "*data:image/jpeg;base64,<base64 encoded image data>*" • "File": image file address, i.e. "*file:///data/user/0/com.abbyy.rtr.reactnative.sample.coreapi/files/pages/page_848b121d-5a7a-4ead-94e6-dd91bc6bfead.jpg*". '*content://*' schema is also supported for Android. Required parameter.
result	object	Result image description, represented by the following parameters: • **compressionLevel** (string): the uniform image compression scale for lossy formats. Valid for export to JPG only. This parameter can be set to the following values: • 'Low' • 'Normal' • 'High' • 'ExtraHigh' Default: 'Low'. • **exportType** (string): result image will be saved to this format. This parameter can be set to the following values: ○ 'Jpg' ○ 'Png' Default: "Jpg". • **destination** (string): result image will be saved to corresponding file ("File") or returned as encode base64 image string ("Base64"). Default: "Base64". • **filePath** (string): full path to the exported file. This parameter is present if the destination is 'File'.

Result

This section describes the object that represents image capture results. Returned parameters depend on the scenario.

Parameter	Value type	Description
imageUri	string	Exported image returned as URI. This image corresponds to the description in the passed **result** parameter.
imageSize	object	Exported image size represented by its width and height in pixels. The values of the width and height are stored in the following parameters: • **width** (int) • **height** (int)

Example of a result JSON.

```
{
  "imageUri": "file://data/user/0/com.abbyy.rtr.cordova.sample/files/
{guid}.jpg",
  "imageSize": {
    "width": 1966,
   "height": 3474
  }
}
```

exportImagesToPdf method

Exports images to PDF format. An array of images can be exported, see options description for details.

```
export async function exportImagesToPdf(settings)
```

Parameters

settings

Object specifying parameters of the image export to PDF (see Options).

Return values

The method returns result depending on how the scenario finished (see Result).

Options

The table below describes parameters that you can pass as the *settings* argument to change image export to PDF settings. Omitting a parameter means that a default setting will be used.

Parameter	Value type	Description
licenseFileName	string	The name of the license file. This file must be located in the **/assets/** directory in your project. Default: "MobileCapture.License".
images	object[]	Array of images for export with corresponding information. Each image can have the following parameters: • **pageSize** (object): image size represented by its width and height in points (1/72 of an inch). I.e., a page size of A4 is 595x842. If both width and height are set to 0, image size will be detected in pixels. The values of the width and height are stored in corresponding width and height parameters: • **width** (int) • **height** (int) • **compressionLevel** (string): the uniform image compression scale for lossy formats. Valid for export to JPG only. This parameter can be set to the following values: • 'Low' • 'Normal' • 'High' • 'ExtraHigh' Default: 'Low'. • **imageUri** (string): Image source for the operation passed as URI. This parameter can be set to the following values: • "Base64": you can pass an encode base64 image string, i.e. "*data:image/jpeg;base64,<base64 encoded image data>*" • "File": image file address, i.e. "*file:///data/user/0/com.abbyy.rtr.reactnative.sample.coreapi/files/pages/page_848b121d-5a7a-4ead-94e6-dd91bc6bfead.jpg*". '*content://*' schema is also supported for Android. Required parameter.
result	object	Result PDF file description, represented by the following parameters: • **destination** (string): result image will be saved to corresponding file ("File") or returned as encode base64 image string ("Base64"). Default: "Base64". • **filePath** (string): full path to the exported file. This parameter is used only if the **destination** is 'File'. Otherwise the value will be ignored. In case the

Parameter	Value type	Description
		destination is 'File' and this parameter is not set, the path will be generated automatically.
pdfInfo	object	Extra information corresponding to the PDF file. This object can have the following parameters: • **title** (string) • **subject** (string) • **keywords** (string) • **author** (string) • **company** (string) • **creator** (string) • **producer** (string)

Result

This section describes the object that represents image capture results. Returned parameters depend on the scenario.

Parameter	Value type	Description
pdfUri	string	Result URI, corresponding to the description in the passed **result** parameter. If the PDF file destination was 'File', this parameter value corresponds to the **filePath** parameter value with '*file://*' prefix.

Example of a result JSON.

```
{
  "pdfUri": "file://data/user/0/com.abbyy.rtr.cordova.sample/files/
{guid}.pdf"
}
```

Copyright and Trademark Notices

ABBYY Mobile Capture © 2020 ABBYY Development, Inc.
ABBYY is a registered trademark or a trademark of ABBYY Software Ltd.

AndroidPdfViewer

Copyright 2017 Bartosz Schiller

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Kotlin

Copyright 2010-2018 JetBrains s.r.o.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

react

MIT License

Copyright (c) Facebook, Inc. and its affiliates.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-native-keyboard-aware-scroll-view

The MIT License (MIT)

Copyright (c) 2015 APSL

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

React Native

MIT License
Copyright (c) Facebook, Inc. and its affiliates.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

React Native PDFView

MIT License
Copyright (c) 2018 Maksym Rusynyk

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

React Native ActionSheet

The MIT License (MIT)
Copyright (c) 2019 Talut TASGIRAN

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

hamcrest

BSD License

Copyright (c) 2000-2015 www.hamcrest.org
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Hamcrest nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

jacoco-android-plugin

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

jetpack

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution." "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

JUnit

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM

CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the

terms of this Agreement, and without further action by the parties hereto, such
provision shall be reformed to the minimum extent necessary to make such
provision valid and enforceable.

If Recipient institutes patent litigation against any
entity (including a cross-claim or counterclaim in a lawsuit) alleging that the
Program itself (excluding combinations of the Program with other software or
hardware) infringes such Recipient's patent(s), then such Recipient's rights
granted under Section 2(b) shall terminate as of the date such litigation is
filed.

All Recipient's rights under this Agreement shall terminate if it fails to
comply with any of the material terms or conditions of this Agreement and does
not cure such failure in a reasonable period of time after becoming aware of
such noncompliance. If all Recipient's rights under this Agreement terminate,
Recipient agrees to cease use and distribution of the Program as soon as
reasonably practicable. However, Recipient's obligations under this Agreement
and any licenses granted by Recipient relating to the Program shall continue
and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in
order to avoid inconsistency the Agreement is copyrighted and may only be
modified in the following manner. The Agreement Steward reserves the right to
publish new versions (including revisions) of this Agreement from time to time.
No one other than the Agreement Steward has the right to modify this Agreement.
The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to
serve as the Agreement Steward to a suitable separate entity. Each new version
of the Agreement will be given a distinguishing version number. The Program
(including Contributions) may always be distributed subject to the version of
the Agreement under which it was received. In addition, after a new version of
the Agreement is published, Contributor may elect to distribute the Program
(including its Contributions) under the new version. Except as expressly stated
in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to
the intellectual property of any Contributor under this Agreement, whether
expressly, by implication, estoppel or otherwise. All rights in the Program not
expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the
intellectual property laws of the United States of America. No party to this
Agreement will bring a legal action under this Agreement more than one year
after the cause of action arose. Each party waives its rights to a jury trial
in any resulting litigation.

react-native-fs

The MIT License (MIT)

Copyright (c) 2015 Johannes Lumpe

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-native-share

The MIT License (MIT)

Copyright (c) 2015 Esteban Fuentealba

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

react-native-svg

The MIT License (MIT)

Copyright (c) [2015-2016] [Horcrux]

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

robolectric

The MIT License

Copyright (c) 2010 Xtreme Labs, Pivotal Labs and Google Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

unmock plugin

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity

exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2018 Björn Quentin

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Portions of this software are derivative works based on software developed by ABBYY Production LLC, Copyright (c) 2019 ABBYY Production LLC. All rights reserved.

All other trademarks and copyrights are the property of their respective owners.