**Image Conversion Module**

**Description**

Welcome to Image Conversion Module. This module can be used in Mendix apps to perform various conversion operations on images based on the different java actions. The module contains the four custom java actions which help users to convert, resize as well as add watermarks on the images.

**Typical Usage Scenario**

This module can be used in Mendix apps to perform various conversion operations on images based on the different java actions

**PreRequisites**

NA

**Features and Limitations**

**Features**

So now we will tell you how you can use Java Actions to perform various Image Conversions.

1. Java\_ResizeImage: This Java Action is used to resize the image based on the 4 input parameters which are Input Image Path, Output Image Path, Width of the Image, and Height of the Image. So, the user needs to pass these four parameters in their microflow to this Java Action Call so based on the inputs passed by the user a new image gets created in the output folder based on the width and height given by user. In return Java Action will send the Boolean value **True** to let you know whether the image is resized or not.
2. JavaAction\_ConvertImage\_Into\_Different\_Formats: This Java Action is used to convert the image based on the 3 input parameters which are Input Image Path, Output Image Path, and Image Format. In this Java Action user can convert their images from one format to another. So, the user needs to pass three parameters in their microflow to this Java Action Call so based on the inputs passed by the user a new image gets created in the output folder based on the format given by the user. In return Java Action will send the Boolean value **True** to let you know whether the image is converted or not.
3. JavaAction\_AddWaterMark\_Basedon\_InputText: This Java Action is used to add the Watermark in the image based on the 3 input parameters which are Input Image Path, Output Image Path, and Input Text. In this Java Action user can add his own Watermark into the input images based on the input text which used as Watermark on the image and create a new image in the output path provided by user. In return Java Action will send the Boolean value **True** to let you know whether the Watermark has been added to the image or not.
4. JavaAction\_AddWaterMark\_Basedon\_InputImage: This Java Action is used to add the Watermark in the image based on the 3 input parameters which are Input Image Path, Output Image Path, and Input Image. In this Java Action user can add his own Watermark into the input images based on the input image which used as Watermark on the selected image and create a new image in the output path provided by user. In return Java Action will send the Boolean value **True** to let you know whether the Watermark has been added to the image or not.

**Limitations**

1. In this image conversion module, we are not converting any image into grayscale, we are just converting its format. (Example: PNG – JPG or JPG - PNG).
2. In this image conversion module, we are not performing crop operation in the images, we are just performing resizing operation on the images based on the width and height specified by user.
3. In this image conversion module, we can add Watermark in the images, but we cannot remove any Watermark from any image.

**Dependencies**

1. This module works in Mendix Version 9.18.0 and higher.
2. Community Commons module is required for the example which is present in the module.

**Installation**

1. Download the Image Conversion Module from the Mendix App Store.
2. In Docs folder, you’ll get the information about the Java Actions which helps you to perform different image conversions.
3. In Java Actions folder, you’ll get all the Java Actions which you need to perform your operations.
4. In Example Folder, we provided you with the example by creating a sample page and microflows which helps you understand Java Actions.

**Configuration: NA**

**Category: Module**

**Subcategory: All**

**Visibility: Public Marketplace**

**Studio Pro Version: 9.18.0**