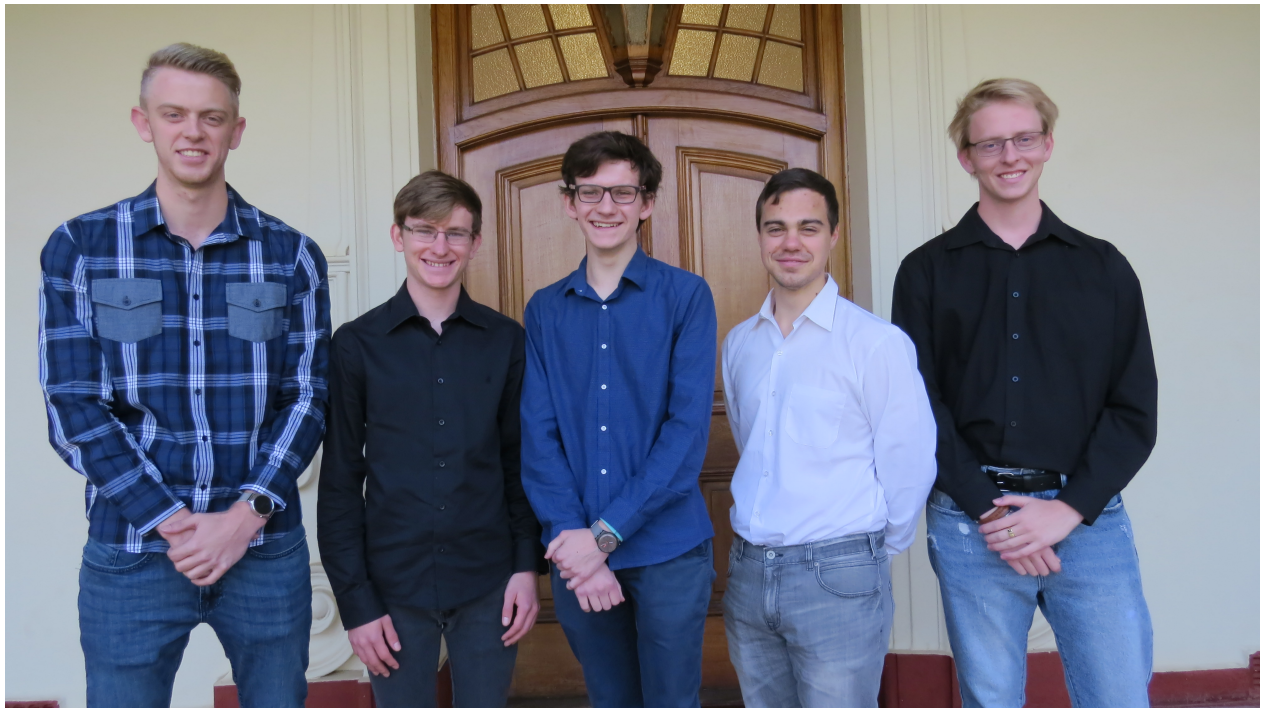


UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

User Manual

Team: Java the Hutts
2017



Nicolai van Niekerk
nicvaniek@gmail.com

Marno Hermann
marno@barnton-consulting.co.za

Stephan Nell
nellstephanj@gmail.com

Jan-Justin van Tonder
J.vanTonder@tuks.co.za

Andreas Nel
nel.andreas1@gmail.com



Contents

1	Product Description	1
2	API Endpoints	1
2.1	Extract Text	1
2.2	Extract Face	2
2.3	Extract All	3



1 Product Description

Hutts Validation is a WebAPI that serves as an electronic ID verification system. The purpose of the system is to extract, process and validate information from a photo of an ID book, ID card or driver's license. Given a photo of the form of identification as well as given personal information, including the name, surname and a photo of the individual's face, the system will return a percentage match score for each element. This is done by validating the given information against extracted fields from the photo of the identification document. Additionally, the system can return the extracted fields given the photo of the form of identification.

2 API Endpoints

This section aims to give a detailed explanation of each endpoint to give the user a clear understanding of how to make a call to it, as well as what to expect from responses.

2.1 Extract Text

1. **Description:** Extracts textual information from the photo of the ID and returns it.
2. **URL:** */extractText*
3. **Method:** POST
4. **URL Parameters:** None
5. **Data Parameters:**

```
{  
  "image" : [blob]  
}
```

Example:

```
{  
  "image" : idDocument.jpg  
}
```

6. **Success Response:**

- **Code:** 200
- **Content:**

```
{  
  "ExtractedFields": {  
    "Surname": "Doe",  
    "Names": "John Jane",  
    "Sex": "M",  
    "Nationality": "RSA",  
    "Identity_Number": "6944585228083",  
    "Date_of_Birth": "06-05-1996",  
    "Country_of_Birth": "RSA",  
    "Status": "Citizen"  
  }  
}
```

7. **Error Response:** TODO



8. Sample Call:

```
$.ajax({
  type: "POST",
  url: "http://localhost:5000/extractText",
  data: {
    "image" : idDocument.jpg
  },
  success: function(data){
    console.log(data);
  }
});
```

9. Notes and Comments:

2.2 Extract Face

1. **Description:** Extracts the face from the photo of the ID and returns it.

2. **URL:** */extractFace*

3. **Method:** POST

4. **URL Parameters:** None

5. **Data Parameters:**

```
{
  "image" : [blob]
}
```

Example:

```
{
  "image" : idDocument.jpg
}
```

6. **Success Response:**

- **Code:** 200
- **Content:**

```
{
  "ExtractedFace": "face.jpg"
}
```

7. **Error Response:** TODO

8. Sample Call:

```
$.ajax({
  type: "POST",
  url: "http://localhost:5000/extractFace",
  data: {
    "image" : idDocument.jpg
  },
  success: function(data){
    console.log(data);
  }
});
```



9. Notes and Comments:

2.3 Extract All

1. **Description:** Extracts textual information and face from the photo of the ID and returns it.
2. **URL:** */extractAll*
3. **Method:** POST
4. **URL Parameters:** None
5. **Data Parameters:**

```
{  
  "image" : [blob]  
}
```

Example:

```
{  
  "image" : idDocument.jpg  
}
```

6. Success Response:

- **Code:** 200
- **Content:**

```
{  
  "ExtractedData": {  
    "Surname": "Doe",  
    "Names": "John Jane",  
    "Sex": "M",  
    "Nationality": "RSA",  
    "Identity_Number": "6944585228083",  
    "Date_of_Birth": "06-05-1996",  
    "Country_of_Birth": "RSA",  
    "Status": "Citizen",  
    "Face": face.jpg  
  }  
}
```

7. Error Response: TODO

8. Sample Call:

```
$.ajax({  
  type: "POST",  
  url: "http://localhost:5000/extractAll",  
  data: {  
    "image" : idDocument.jpg  
  },  
  success: function(data){  
    console.log(data);  
  }  
});
```

9. Notes and Comments:

