|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Request Method | Value | E.g Input | E.g Output | Security |
| Get | /hello | - | Hoila | None |
|  | /checkUsername/{username} | - | {  "success": false  } | None |
| Get | /degrees | - | List of all Degrees |  |
| Get | /degrees/{degreeId} | - | {  "id": "2",  "firstName": "Rahjabeen",  "lastName": "UmerSheikh",  "degreeType": "MS",  "universityId": "10"  } |  |
| Post | /degrees | JSON FORMAT of  Degree("2","Rahjabeen","Abbasi","MS") | Status: 201 Created |  |
| Put | /degrees | JSON FORMAT of  Degree("2","Rahjabeen","Abbasi","MS") | - |  |
| Delete | /degrees/{degreeId} |  | - |  |
| Post | /token/generate-token | {  "username":"giki",  "password": "123"  } | {  "token": "xxxx "  } | None |
| Post | /signup | { "username":"lums", "password":"abc",  "roles": [  "USER"  ], "email":"admin@lums.edu.pk",  "active":"true", "name":"Lahore University of M&S"  } | Status: 201 Created | None |
| Get | /users |  | List of all registered users(univsersities) along with their details |  |
|  |  |  |  |  |
| Get | /users/{usersId} | /users/1 | {  "id": "1",  "username": "giki",  "password": "$2a$04$JPQDDreU05bkZmZNGKUp8u5xYese3mQDf7po.6sLoV.QuMv4F2H8C",  "roles": [  "USER"  ],  "email": "admin@giki.edu.pk",  "active": "true",  "name": "Ghulam Ishaq Khan Institute"  } |  |
| Get | /fixedFormFields |  | [  {  "name": "StudentName",  "validators": [  {  "id": "2",  "name": "alphaReq",  "regex": "[a-zA-z]+",  "type": "text"  }  ],  "customError": "Username is incorrect",  "attributeType": "String"  },  {  "name": "GPA",  "validators": [  {  "id": "8",  "name": "floatReq",  "regex": "[0-9]\*[.]?[0-9]+",  "type": "number"  }  ],  "customError": "GPA is incorrect",  "attributeType": "Number"  },  {  "name": "DegreeType",  "validators": [  {  "id": "2",  "name": "alphaReq",  "regex": "[a-zA-z]+",  "type": "text"  }  ],  "customError": "DegreeType is incorrect",  "attributeType": "String"  },  {  "name": "DegreeName",  "validators": [  {  "id": "1",  "name": "alpha",  "regex": "[a-zA-z]\*",  "type": "text"  }  ],  "customError": "DegreeName is incorrect",  "attributeType": "String"  }  ] |  |
| Get | /fixedFormFields/{fieldName} | /fixedFormFields/DegreeType | {  "name": "DegreeType",  "validators": [  {  "id": "2",  "name": "alphaReq",  "regex": "[a-zA-z]+",  "type": "text"  }  ],  "customError": "DegreeType is incorrect",  "attributeType": "String"  } |  |
| Get | /universities/{universityId}/formFields | /universities/10/ formFields | {  "id": "10",  "name": null,  "formFields": null,  "firstTime": null  } |  |
| Get | /universities/{universityId}/degrees | /universities/10/degrees | [  {  "id": "2",  "firstName": "Rahjabeen",  "lastName": "UmerSheikh",  "degreeType": "MS",  "universityId": "10"  },  {  "id": "4",  "firstName": "Musab",  "lastName": "Hameed",  "degreeType": "MBBS",  "universityId": "10"  }  ] |  |
| Get | /universities/{universityId}/degrees/{degreeId} | localhost:8090/universities/654645/degrees/3 | {  "id": "3",  "firstName": "Urwah",  "lastName": "QA-Engineer",  "degreeType": "BS",  "universityId": "654645"  } |  |
| Get | localhost:8090/universities/names |  | [  {  "id": "10",  "name": "GIKI",  "formFields": null,  "firstTime": null  },  {  "id": "20",  "name": "LUMS",  "formFields": null,  "firstTime": null  },  {  "id": "30",  "name": "NUST",  "formFields": null,  "firstTime": null  }  ] |  |
| Get | /universities/names |  | [  {  "id": "10",  "name": "GIKI",  "formFields": null,  "firstTime": "True"  },  {  "id": "20",  "name": "LUMS",  "formFields": null,  "firstTime": "True"  },  {  "id": "30",  "name": "NUST",  "formFields": null,  "firstTime": "False"  }  ] |  |
| Get | /universities |  | [  {  "id": "10",  "name": "GIKI",  "formFields": null,  "firstTime": "True"  },  {  "id": "20",  "name": "LUMS",  "formFields": null,  "firstTime": "True"  },  {  "id": "30",  "name": "NUST",  "formFields": null,  "firstTime": "False"  }  ] |  |
| Get | /universities/{universitiesId} | /universities/30 | {  "id": "30",  "name": "NUST",  "formFields": null,  "firstTime": "False"  } |  |
| Post | /universities | {  "id": "40",  "name": "IBA",  "formFields": null,  "firstTime": "True"  } |  |  |
| Put | /universities | {  "id": "40",  "name": "Not IBA",  "formFields": null,  "firstTime": "True"  } |  |  |
| Delete | /universities/{universitiesId} | /universities/40 |  |  |
| Get | /validators |  |  | None |
| Get | /validators/{validatorId} |  |  | None |
| Post | /validators | Id: name: regex: type:  ("1","alpha", "[a-zA-z]\*","text") |  | None |
| Put | /validators |  |  | None |
| Delete | /validators/{validatorId} |  |  | None |

# Security

.antMatchers("/hello").permitAll()

.antMatchers("/universities","/universities/\*").permitAll()

.antMatchers("/validators","/validators/\*").permitAll()

.antMatchers("/token/\*", "/signup" ).permitAll()

# .anyRequest().authenticated()

# Classes

## Degree

**private** String id;

**private** String firstName;

**private** String lastName;

**private** String degreeType;

**private** String universityId;

## Users

**private** String id;

**private** String username;

**private** String password;

**private** List<String> roles;

**private** String email;

**private** String active;

**private** String name;

## FormField

**private** String name; //name of form attributes. studentName, GPA

**private** List<Validator> validators;

**private** String customError;

**private** String attributeType;

## LoginUser

**private** String username;

**private** String password;

## Users

**private** String id;

**private** String username;

**private** String password;

**private** List<String> roles;

**private** String email;

**private** String active;

**private** String name;

**private** String number;

**private** String address;

**private** String representativeName;

**private** String representativeNumber;

**private** String representativeCNIC;

## University

**private** String id;

**private** String name;

**private** List<FormField> formFields;

**private** String firstTime;

## Validator

p**rivate** String id;

**private** String name;

**private** String regex;

**private** String type;