**CAPMATCH SERVER ENDPOINTS**

**Note:***The base path of the API will change depending on the hosting of the backend. It is, therefore, a good idea to keep the base path as a separate variable if Http paths are to be explicitly constructed in the client.*

*For the most part, however, constructing paths will not be needed as the backend implements the* ***HATEOAS*** *(Hypermedia As The Engine Of Application State) paradigm. As a result, resources returned from the server would be embedded with links the define interactions with that object. This makes client code less fragile and makes it possible to change the backend without affecting the client.*

*Currently, the basepath of the staged backend is* <https://capmatch-stage.cfapps.io/>. *Although all the resources are exposed now, I will restrict access to some in due course and will give you the instruction on how to access them if they are needed for the client. The only thing that will change is some Authentication headers would have to be added to the request. Nothing deep*

1. To get a quick overview of the resources available and obtain that path to access the resources, you can send a GET request to the base path. The response will have links to access all exposed resources. An example of the response you will see is shown below.

{

  "\_links": {

    "majors": {

      "href": "https://capmatch-staging.cfapps.io/majors"

    },

    "users": {

      "href": "https://capmatch-staging.cfapps.io/users"

    },

    "faculties": {

      "href": "https://capmatch-staging.cfapps.io/faculties"

    },

    "userPermissions": {

      "href": "https://capmatch-staging.cfapps.io/userPermissions"

    },

    "departments": {

      "href": "https://capmatch-staging.cfapps.io/departments"

    },

    "loginProfiles": {

      "href": "https://capmatch-staging.cfapps.io/loginProfiles"

    },

    "accountConfirmations": {

      "href": "https://capmatch-staging.cfapps.io/accountConfirmations"

    },

    "sDGs": {

      "href": "https://capmatch-staging.cfapps.io/sDGs"

    },

    "interestCategories": {

      "href": "https://capmatch-staging.cfapps.io/interestCategories"

    },

    "students": {

      "href": "https://capmatch-staging.cfapps.io/students"

    },

    "interests": {

      "href": "https://capmatch-staging.cfapps.io/interests"

    },

    "profile": {

      "href": "https://capmatch-staging.cfapps.io/profile"

    }

  }

}

As shown above, to access any resource, you just append the plural form of the resource to the basepath. For example, all majors can be accessed at *<basepath>/majors,* and departments can be accessed at *<basepath>/departments*.

***SECURITY***

The server implements basic authentication using Spring Security. Some routes such as *“<basepath>/accountConfirmations”, “…/loginProfiles”, “…/profile”,* and *“…/userPermissions”* have been completely blocked off as access to these resources would be a security flaw.

Endpoints for login and signup can be accessed without need for authentication. Also, publicly available data such as departments, majors, SDGs, and general interests have been made accessible without need for authorization.

Again, all endpoints for student and faculty data require authorization. This implies that to access any endpoint that starts out with *“<basepath>/users”*, *“…/students”*, or *“…/faculty”* require that the request is authenticated. Both students and faculty are allowed to access these endpoints.

Finally, all endpoints that start out with *“<basepath>/admin”* require authorization to ensure that the request is being made by a designated admin.

**How To**

To enable a request to be authenticated, an *“****Authorization****”* header field needs to be provided. The value of this header field should be a string follows:

Basic **<*email>:<password>***

With the part in bold base64 encoded. **Note** that the password is not plaintext but rather the client-side encoded password.

***SIGNING UP***

To sign up, send a POST request with the signup details to *<basepath>/signup/student* or *<basepath>/signup/faculty* for students and faculty respectively. The body of the POST request must be a JSON object as shown below. For students, a major object must be included, and for faculty a department object can be included. (You can find how to get the majors and departments available above)

To signup a student, the request body would be as shown below

{

"firstname": "Rahul",

"lastname": "Srinivas",

"email": "rahul.srinivas@ashesi.edu.gh",

"password": "SOmeR@ndOmPass",

"bio": "I an amazing young man with utmost passion for learning. My strengths are in everything Computer Science and I am especially good with Web Development",

"major": {

"majorCode": "CS",

"name": "Computer Science",

"\_links": {

"self": {

"href": "https://capmatch-staging.cfapps.io/majors/1"

},

"major": {

"href": "https://capmatch-staging.cfapps.io/majors/1"

},

"students": {

"href": "https://capmatch-staging.cfapps.io/majors/1/students"

},

"department": {

"href": "https://capmatch-staging.cfapps.io/majors/1/department"

}

}

}

}

If the request is successful, you will receive a response with status code *201 (CREATED)*. The response body would also contain a user object which can be used for the rest of the signup process. In this example, the following response is received.

{

    "userId": 1,

    "firstname": "Rahul",

    "lastname": "Srinivas",

    "email": "rahul.srinivas@ashesi.edu.gh",

    "password": "$2a$10$5J0vGXv0snFrCUJMzolz3.6un.HXFSMt8i30Xg77jz/fIyKnDyaJ.",

"bio": "I an amazing young man with utmost passion for learning. My strenths are in everything Computer Science and I am esspecailly good with Web Development,

    "accountStatus": "UNVERIFIED",

    "registrationDate": "2020-05-12T19:58:55.715+0000",

    "permissions": [

        {

            "id": 1,

            "name": "STUDENT"

        }

    ],

    "\_links": {

        "self": {

            "href": "https://capmatch-staging.cfapps.io/users/1"

        },

        "interests": {

            "href": "https://capmatch-staging.cfapps.io/users/1/interests"

        },

        "SDGs": {

            "href": "https://capmatch-staging.cfapps.io/users/1/SDGs"

        },

        "major": {

            "href": "https://capmatch-staging.cfapps.io/students/1/major"

        },

        "supervisor": {

            "href": "https://capmatch-staging.cfapps.io/students/1/supervisor"

        },

        "favouriteSupervisors": {

            "href": "https://capmatch-staging.cfapps.io/students/1/favouriteSupervisors"

        },

        "addInterests": {

            "href": "https://capmatch-staging.cfapps.io/users/1/addInterests"

        },

        "addInterest": {

            "href": "https://capmatch-staging.cfapps.io/users/1/addInterest"

        },

        "addSDGs": {

            "href": "https://capmatch-staging.cfapps.io/users/1/addSDGs"

        },

        "addSDG": {

            "href": "https://capmatch-staging.cfapps.io/users/1/addSDG"

        },

        "sendConfirmation": {

            "href": "https://capmatch-staging.cfapps.io/users/1/sendConfirmation"

        }

    }

}

In keeping with the spirit of HATEOAS, you’ll realize the response also contains the links that will be needed for the rest of the signup process.

Currently, the following constraints apply for signing up:

* Email must be an Ashesi email (i.e. must end in @ashesi.edu.gh).
* Password must be at least 8 characters long.
* All fields must be provided. *Firstname, lastname, email, password* as well as *major* for students and *department* for faculty signups
* The bio field may be added but is not required.

If any of these are violated, you will receive a *406 Not Acceptable* status code from the server.

If a signup is attempted with an existing email, you will receive a *226 IM Used* status from the server.

**Selecting Interests**

Accessing the interests has already been described earlier in the doc.

To save the choices a user makes, you’ll have to send a POST or PUT request to the *“addInterests”* link provided in the response after signing up the student. The body of this response should be a list of all the Interest Objects that were selected by the user. An example request body is shown below.

[

{

           "interestId": 11,

           "name": "Web Technologies",

           "description": "Build the next generation of tech to power the web",

           "\_links": {

               "self": {

                   "href": "https://capmatch-staging.cfapps.io/interests/11"

               },

               "category": {

                   "href": "https://capmatch-taging.cfapps.io/interests/11/category"

               },

               "interestedUsers": {

                   "href": "https://capmatch-staging.cfapps.io/interests/11/interestedUsers"

               }

           }

       },

       {

           "interestId": 21,

          "name": "Software Engineering",

           "description": "Create tools to aid software design and implementation",

           "\_links": {

               "self": {

                   "href": "https://capmatch-staging.cfapps.io/interests/21"

               },

               "category": {

                   "href": "https://capmatch-staging.cfapps.io/interests/21/category"

               },

               "interestedUsers": {

                   "href": "https://capmatch-staging.cfapps.io/interests/21/interestedUsers"

               }

           }

       },

       {

           "interestId": 31,

           "name": "Internet of Things",

           "description": "Connect every device to a single ecosystem",

           "\_links": {

              "self": {

                   "href": "https://capmatch-staging.cfapps.io/interests/31"

               },

               "category": {

                   "href": "https://capmatch-staging.cfapps.io/interests/31/category"

               },

               "interestedUsers": {

                   "href": "https://capmatch-staging.cfapps.io/interests/31/interestedUsers"

               }

           }

       }

    }

]

Once this request is successful, you’ll receive a *200 OK* response from the server.

**Selecting SDGs**

This is similar to how interests are selected. This time though, the request is sent to the *“addSDGs”* link provided in the response after signing up.

**Sending Confirmation/Verification Email**

To send confirmation email, send a POST request to the *“sendConfirmation”*  link provided in the response after signing up.

This endpoint will return a status code of 404 *NOT\_FOUND* if the user specified could not be found, *200 OK* if the user has already been confirmed and not email is sent, or *201 Created* if the email was actually sent.

**Sending Confirmation/Verification Email given an Email**

Alternatively, you can request a confirmation email to be sent by sending a POST request to *<basepath>/users/sendConfirmation*. In this case, the POST request body must be a JSON object with an email field as shown below.

{

"email": "rahul.srinivas@ashesi.edu.gh"

}

This endpoint returns *404 Not Found*  if the email provided isn’t tied to any registered account, and returns the same status codes as the endpoint directly above under the said conditions.

**LOGGING IN**

To login, you send a GET request to *<basepath>/login*. The request body should be a JSON object as shown below.

{

"email": "rahul.srinivas@ashesi.edu.gh",

"password": "SOmeR@ndOmPass"

}

The response will have a status code of *302 Found* if the credentials are correct and will have a body with a JSON object representing the user. The returned object would also contain links for all the actions a user is able to do for use by the client. A sample response is shown below.

{

    "userId": 1,

    "firstname": "Rahul",

    "lastname": "Srinivas",

    "email": "rahul.srinivas@ashesi.edu.gh",

    "password": "$2a$10$5J0vGXv0snFrCUJMzolz3.6un.HXFSMt8i30Xg77jz/fIyKnDyaJ.",

    "accountStatus": "UNVERIFIED",

    "registrationDate": "2020-05-12T19:58:55.000+0000",

    "permissions": [

        {

            "id": 1,

            "name": "STUDENT"

        }

    ],

    "\_links": {

        "self": {

            "href": "https://capmatch-staging.cfapps.io/users/1"

        },

        "interests": {

            "href": "https://capmatch-staging.cfapps.io/users/1/interests"

        },

        "SDGs": {

            "href": "https://capmatch-staging.cfapps.io/users/1/SDGs"

        },

        "major": {

            "href": "https://capmatch-staging.cfapps.io/students/1/major"

        },

        "supervisor": {

            "href": "https://capmatch-staging.cfapps.io/students/1/supervisor"

        },

        "favouriteSupervisors": {

            "href": "https://capmatch-staging.cfapps.io/students/1/favouriteSupervisors"

        },

        "addInterests": {

            "href": "https://capmatch-staging.cfapps.io/users/1/addInterests"

        },

        "addInterest": {

            "href": "https://capmatch-staging.cfapps.io/users/1/addInterest"

        },

        "addSDGs": {

            "href": "https://capmatch-staging.cfapps.io/users/1/addSDGs"

        },

        "addSDG": {

            "href": "https://capmatch-staging.cfapps.io/users/1/addSDG"

        },

        "sendConfirmation": {

            "href": "https://capmatch-staging.cfapps.io/users/1/sendConfirmation"

        }

    }

}

You’ll realize from the response above that the user’s account status is *“UNVERIFIED”*. The account status can be *“****UNVERIFIED***, ***ACTIVE***, ***BLOCKED***, ***EXPIRED****”.* In this case, where the user is yet to verify their account, you can see the response also includes a link where a POST request can be sent to resend the confirmation email to the user.

A user should only be let in when the account status is ***ACTIVE***.

When the login attempt fails, you will receive A *404 Not Found* status code from the server.

Again, however, after 4 failed attempts to login, the user account will be blocked, and you will receive a *433 Locked* status code from the server.

**FORGOT PASSWORD**

To send a password reset email, send a POST request to *<basepath>/users/forgotPassword*. You’ll need to provide an email in the request body.

{

“email”: “[rahul.srinivas@ashesi.edu.gh](mailto:rahul.srinivas@ashesi.edu.gh)”

}

If the provided email cannot be matched with any registered user, a 406 Not Acceptable status code is received. If the email is found and a status code received, a *201 Created* response is received.