

### Service Endpoints:

- 1) **/api/user/post**-This endpoint is used for creating a new user. User's have to register to have access to the application. This endpoint will interact with the Home page user interface. An example of a user interaction: Adam Admin hasn't registered an account yet, before he can create awards he will have to register.

Example:

POST <http://www.easymanage.com/users>

```
{  
  "id": 0,  
  "Employee_Number": 1000  
  "First_Name": "Will",  
  "Last_Name": "Cowan",  
  "Email": "email@company.com",  
  "Password": "password"  
  "Role": "Analyst"  
}
```

A successful attempt to create a user will result in the user getting a HTTP/ 201 Created message "User Account Successfully Created"

A failed attempt to create a user will result in the user getting a HTTP/409 Conflict message "User Account Already Exists"

- 2) **/api/user/get** – This endpoint will be used to log into the application. It will be any user with an accounts first step to log into the application. Mary Manager will use this endpoint to log into the application to retrieve award reporting information.

Example:

GET <http://www.easymanage.com/users>

```
{  
  "id": 0,  
  "Employee_Number": 1000  
  "First_Name": "Will",  
  "Last_Name": "Cowan",  
  "Email": "email@company.com",  
  "Password": "password"  
  "Role": "Analyst"  
}
```

A successful attempt to login will result in access to the application would result in the user getting a HTTP/200 OK Message "Welcome to EasyManage!"

A failed attempt to login will result in the user not having access to the application and the user getting a HTTP/ 400 Bad Request message "Retry Login Information or create an account"

- 3) /api/award/post** – This endpoint will interact with the Create user interface. Users will use it to create new awards. For example, Adam Admin would use this frequently to create award records. This is a very important endpoint; it will serve as the basis for entering award reporting info in the application.

Example:

POST http://www.easymange.com/create

```
{
  "id": 0
  "Award_Number": 1000,
  "Award_Name": "Test Award",
  "Award_Sponsor": "Maryville University",
  "Award_PI": "John Doe",
  "Dept_Number": 100
  "Start_Date": "10/1/21",
  "End_Date": "10/31/22",
  "Reporting_Period": "Annual Financial Report",
  "Special_Report": "Annual Effort Certification",
  "Report_Status": "Incomplete"
}
```

A successful attempt to create a record will result in a user getting a HTTP/200 OK message "Award Successfully Created"

A failed attempt will result in the user getting a HTTP/400 Bad Request message "Please provide all required information and resubmit"

- 4) /api/award/get** – This service endpoint will be used to retrieve award data. It will interact with the List page. Mary Manager will use it to search for award reporting information. For example, she could search for reporting information for award number 1000.

This will be the most complicated endpoint. Ideally there will be multiple search criteria for users.

The user will be able to select the search criteria. For example the user could select Award Number, Award PI, or Department Number to find relevant awards.

Examples:

GET http://www.easymanage.com/list

```
{  
  "Award_Number": 1000  
}
```

GET http://www.easymanage.com/list

```
{  
  "Award_Pl": "John Doe"  
}
```

GET http://www.easymanage.com/list

```
{  
  "Dept_Number": 100  
}
```

A successful attempt with result in a HTTP/200 OK message "Award(s) Found"

A failed attempt could be for 2 reasons:

1<sup>st</sup> – The criteria doesn't exist; this would result in a HTTP/ 404 Not Found message "Award(s) Not Found"

2<sup>nd</sup>– The criteria isn't formatted correctly; this would result in a HTTP/400 Bad Request message "Bad Criteria Try Again"

- 5) /api/award/put-** This endpoint will be used to update reporting information for an award. It will interact with the Update user interface. Amy Analyst will use this endpoint to update reporting information after she renegotiates a contract. Adam Admin could also use this if he makes an error in his initial award creation.

Example:

PUT http://www.easymanage.com/update

```
{  
  "id": 0  
  "Award_Number": 1000,  
  "Award_Name": "Test Award",  
  "Award_Sponsor": "Maryville University",  
  "Award_Pl": "John Doe",  
  "Dept_Number": 100  
  "Start_Date": "10/1/21",  
  "End_Date": "10/31/23",  
  "Reporting_Period": "Annual Financial Report",  
  "Special_Report": "Annual Effort Certification",  
  "Report_Status": "WIP"  
}
```

- 6) api/comment/create – This award will allow users to create a comment for an award. That way users do not have to email each other for routine work updates. Alice Analyst could leave a comment on an award like, “Waiting on Department”.

Example:

```
POST http://www.easymanage.com/list
{
  "id": 1
  "Employee_Number": 2000
  "Award_Number": 1500
  "Comments": "Report is ready, please submit"
}
```

- 7) api/comment/get – This endpoint will retrieve the comments for an award. Mary Manager could use it to see if Alice Analyst had left any comments about her work. This would help keep Mary in the loop without relying on emails.

Example:

```
GET http://www.easymanage.com/list
{
  "Award_Number": 1500
}
```

- 8) api/notification/get – This is my stretch service endpoint. It will provide users with real time notifications regarding award report status changes. For example, if a report status is updated to “Review Completed Report Signed” Alice Analyst will get a notification and know to submit the report to the sponsor.

```
GET http://www.easymanage.com/home
{
  "Employee_Number": 2000
  "Award_Number": 1500
  "Report_Status": Complete Ready for Review"
}
```

# USER INTERFACE ENPOINT DIAGRAM:

