



ITERATION 3

DEVELOPER PORTAL TEAM

ANDREW STAFFORD, AUSTIN BELT, JAKE CHAPPELL, JAREN PROVOST

MENTOR FEEDBACK

“YOU GUYS ARE DOING REALLY GOOD” -RYLAND

Mid Iteration Feedback:

- Suggested we have our data loader be triggered by a button (our app will initially only be used by Accutech staff)

Adjustments Made:

- The home page now contains a button to trigger the data loader

End of Iteration Feedback:

- Gave some guidance on sessions, and suggested we have a meeting with the staff assigned to Cheetah Go

Adjustments Made:

- We will consider having a meeting with the Cheetah Go team to get a better understanding of how they do their user login sessions

CLIENT FEEDBACK

“IT’S LOOKING GOOD FELLAS” -CAM

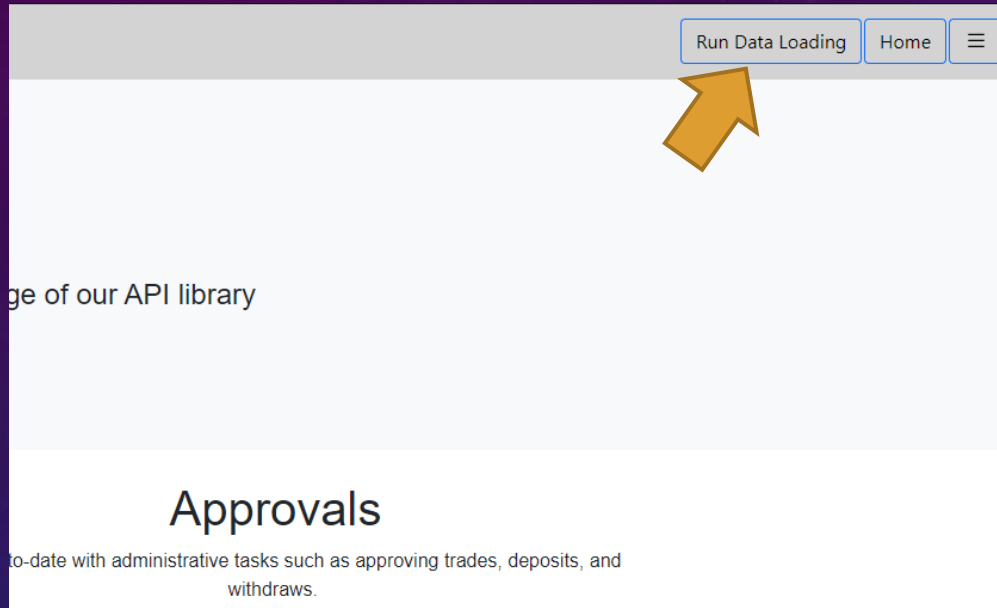
Feedback:

- Cam suggested changing the DB structure to fit Accutech’s schema and provided a model
- Cam suggested making the different versions of our reference page feature a singular endpoint rather than a collection of endpoints


Adjustments Made:

- Rewrote our DB schema to fit Cam’s model and optimize data loading
- Adjusted our sidebar to have collections drop down with endpoints, and adjusted our references page to feature one endpoint’s information

FUNCTIONALITY 1 – DATA LOADER



Clicking the “Run Data Loading” button will load data in from Accutech’s Cheetah swagger page and check for updates if there are any.



	EndpointId	EndpointName	URLSuffix	Description	CollectionId
1	4776	Get account information.	/api/v6/Accounts	NULL	1140
2	4777	Add a new NameAndAddress (Identity Record). Plea...	/api/v6/NameAndAddresses	NULL	1145
3	4778	Get NameAndAddress (Identity Record) information.	/api/v6/NameAndAddresses/{identityRecordId}	NULL	1145
4	4779	Add a new Investment Model	/api/v6/Modeling/InvestmentModel	NULL	1146
5	4780	Get investment models.	/api/v6/Modeling/InvestmentModel	NULL	1146
6	4781	Used to obtain the internal Id values of Cheetah obje...	/api/v6/Lookup	NULL	1147
7	4782	Get Transaction Import Setup Information.	/api/v6/InterfaceMasterTypes/TransactionImportSe...	NULL	1148
8	4783	Get Account Extract Setup Information.	/api/v6/InterfaceMasterTypes/AccountExtractSetups	NULL	1148
9	4784	Get Interface Master Type Information.	/api/v6/InterfaceMasterTypes	NULL	1148
10	4785	Get Installation information.	/api/v6/Installation/InstallationInfo	NULL	1149
11	4786	Get account holding information.	/api/v6/Holdings	NULL	1150
12	4787	Download given file Id.	/api/v6/FileAttachments/{fileAttachmentId}/Downlo...	NULL	1151
13	4788	Get NameAndAddress (Identity Record) information.	/api/v6/NameAndAddresses/{identityRecordId}	NULL	1145
14	4789	Get file attachments.	/api/v6/FileAttachments	NULL	1151
15	4790	Get fee setting schedule information.	/api/v6/FeeSettings/FeeSettingSchedule	NULL	1152
16	4791	Save fee paid from account. This will delete any exist...	/api/v6/FeeSettings/FeePaidFromAccount	NULL	1152
17	4792	Get fee paid from accounts information.	/api/v6/FeeSettings/FeePaidFromAccount	NULL	1152
18	4793	Update a fee setting	/api/v6/FeeSettings/{id}	NULL	1152
19	4794	Add a new fee setting	/api/v6/FeeSettings	NULL	1152

✓ Query executed successfully.

FUNCTIONALITY 2 – DYNAMIC EDITS

Cheetah API References

On this page you'll find documentation on the *Get acc*

Edit

Save

Cancel

GET Get account information.

/api/v6/Accounts

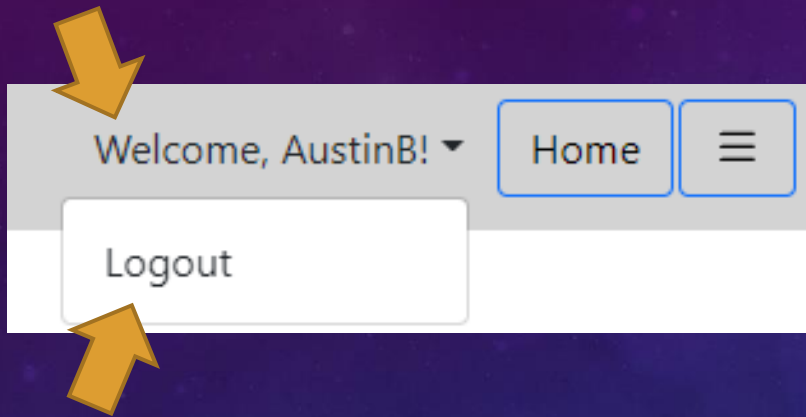
Endpoint Description:

Get information for accounts

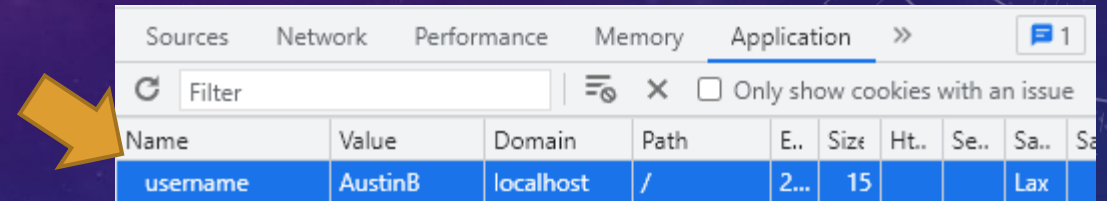
Similar to last iteration, you can make dynamic edits to endpoint descriptions. However, this iteration it can be done right from the References page.

	EndpointId	EndpointName	URLSuffix	Description	CollectionId
1	4776	Get account information.	/api/v6/Accounts	Get information for accounts	1140

FUNCTIONALITY 3 – LOGOUT

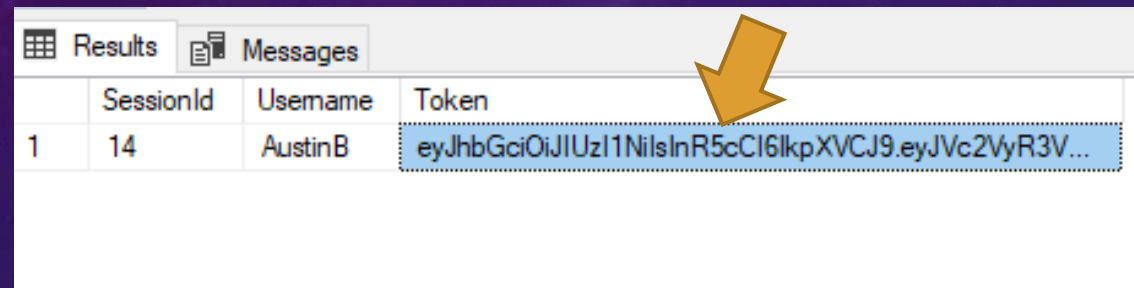


After logging in, the user's Accutech username is displayed on the navigation bar. If the user wishes to log out, clicking their username will reveal a dropdown with a Logout button. After logging out the user will be pushed back to the Login page.



This is made possible by storing the user's username in a cookie. The cookie is dropped upon logging out.

FUNCTIONALITY 4 – STORING USER INFORMATION



	SessionId	Username	Token
1	14	AustinB	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJVc2VyR3V...

After a user logs in, if their username and JWT token have not already been saved, it will be saved to our database. This will be useful in future iterations when the console functionality is implemented.

AESTHETICS

Collections

Account ▾

AccountCategory ▾

AccountGroup ▾

/api/v6/AccountGroups

/api/v6/AccountGroups/{accountGroupId}

/api/v6/AccountGroups/{accountGroupId}/A

The sidebar and references page are now organized by singular endpoints rather than entire collections. The sidebar collections are now a dropdown that show you all the endpoints among that collection

GET Get account information.

POST Add a new Account

PUT Update an Account

DELETE Delete scheduled transaction setups

Each type of HTTP call now has their own stylish icon

4TH ITERATION FEATURES

- **FR6:** The API references page shall have a list of links to objects and endpoints included in the documentation that allows the user to navigate to where that object or endpoint is listed in the documentation [*Medium Priority*] (**BR2**)
 - Finishing the data loader to where it loads in schema references and can track nested schema reference objects and display them on the references page.
- **FR8:** The application shall have an interactive console/sandbox page to test API calls with live data. [*High Priority*] (**BR2**)
 - Sculpting out the console page to the point where it can make calls to Accutech's servers with the correct endpoint and parameters and displays it.
- **FR9:** The console/sandbox page shall have capability to test API calls with live data and responses. [*High Priority*] (**BR2**)
 - Writing back-end methods for the console page to make the API calls to Accutech's servers.
- **FR11:** The administration page shall let administrators create the form for each endpoint that will then be seen in the documentation. [*High Priority*] (**BR2**)
 - Increasing the functionality of dynamic edits, allowing for edits of parameters, responses, etc.
- **FR13:** The application shall automatically update changes made to the Cheetah API by parsing the data from the existing Cheetah API swagger page. [*High Priority*] (**BR2**)
 - Finishing the back-end data loader method to load in schema references and nest them correctly in our database.
- **NR6:** The access of each user shall be session based. [*Medium Priority*] (**BR2**)
 - Finishing the session functionality of our app to where changes will be tracked with user information to avoid overriding edits when loading data.

TEAMWORK

- Jake:
 - Remade database with new structure provided by Cam
 - Wrote back-end logic for the data loader
- Andrew:
 - Wrote front-end logic for sidebar organization
- Jaren:
 - Wrote logic for making edits on the references page
- Austin:
 - Wrote logic for saving user information (including username in site cookies)
 - Wrote logic for logout functionality

RANDOM

KEEPING WITH THE THEME OF ANIMAL PICTURES, THIS ITERATION WE HAVE
JAKE'S DOGS

Josie:

- Hates children
- Loves adults



Dash:

- Closeted alpha male
- Killer underbite



Chip:

- Always happy
- Unhealthily food driven



SOFTWARE!

Things to say:

Yes indeed, all the team members have access to the dev environment!

Things to show:

- Runnable
- Functionality
- Modularity
- Testing
- Clean code

DOCUMENTATION

[User Docs](#)

[Development Docs](#)

[Deployment Docs](#)