BaseSettings

Responsibilities
- Store the configuration for app server such as authentication defaults, database configuration, pagination settings, static file locations and secret keys.
- Different environments can extend this class to modify the configurations.

Collaborators
- None

Parent Class: DjangoDefaults

Child Classes: DevSettings, ProdSettings

Model

Responsibilities
- Abstract the database so we can seamlessly change implementation
- Provide python interface to creating tables
- Manage relationships between tables
- Handle all CRUD operations for respective table

Collaborators: ModelSerializer,

Parent Class: None

Child Classes: SportsCredUser, Post, ACS, Follows, DebatePost Agrees, SocialPost, Likes, QuestionnnaireResponse, Sports, Teams, Player, Highlights, PredictionChoice

FilterSet

Responsibilities
- Handle parsing query string for GET requests
- Define valid query params for models

Collaborators: Respective Model class. (UserFilter collaborates with SportsCredUser model)

Parent Class: None

Child Classes: UserFilter, PredictionFilter, PostFilter, QuestionFilter

ModelSerializer

Responsibilities
- Convert json request body into respective model object
- Convert model object to json to be sent over http
- Define what fields a consumer is able to see from the model

Collaborators
- Model

Parent Class: None

Child Classes: UserSerializer, PostSerializer, PredictionSerializer, SportSerializer, TeamSerializer, PlayerSerializer

ViewSet

Responsibilities
- Define the actions for a set of resources exposed by the app server's API that will be handled by a router class
- Usually map to a Model but not necessarily. Model doesn't have any direct interaction with this class

Collaborators: DefaultRouter

Parent Class: None

Child Classes: UserViewSet, PostViewSet, QuestionsViewSet PredictionViewSet

BasePermission

Responsibilities
- Authorization on end points
- Performs a series of functions based off viewset input to either allow a request or to reject it

Collaborators: ViewSet

Parent Class: None

Child Classes: AnonCreateAndUpdateOwnerOnly

ChatConsumer

Responsibilities
- Connect web socket to front end for chats and notifications
- Alert consumer when a new chat message or notification is received
- close connection when user logs off

Collaborators
- Permissions.IsAuthenticated (permission class)

Parent Class: JsonWebSocketConsumer

Child Classes: None

ACSTask

Responsibilities
- adhoc updates to ACS tasks. Not directly exposed through Web API
- Schedule ACS updates for users. (decay of ACS daily)

Collaborators: None

Parent Class: None

Child Classes: None

SportsCredUser

Post

Responsibilities
- knows its indentifier
- knows passwords/usernames
- knows whether a user is admin
(just a boolean)
- knows which ACS it has for each sport
- knows what posts it's created
- know what predictions it's made
- knows its highlights
- knows the posts it likes for SocialPost
- knows the rating it gave for DebatePost

Collaborators: Post

Parent Class: User (User is a subclass of Model)

Child Classes: None

Responsibilities
- knows its indentifier
- Given a user_id can find all the posts for by user
- knows its title
- knows its content
- knows any attachments

Collaborators:

Parent Class: User (User is a subclass of Model)

Child Classes: DebatePost, SocialPost

DebatePost

Responsibilities
- knows its indentifier
- knows passwords/usernames
- knows whether a user is admin
(just a boolean)
- knows which ACS it has for each sport
- knows what posts it's created
- know what predictions it's made
- knows its highlights
- knows the posts it likes for SocialPost
- knows the rating it gave for DebatePost

Collaborators: Post

Parent Class: User (User is a subclass of Model)

Child Classes: None

SocialPost

Responsibilities
- knows its indentifier
- Given a user_id can find all the posts for by user
- knows its title
- knows its content
- knows any attachments

Collaborators:

Parent Class: User (User is a subclass of Model)

Child Classes: DebatePost, SocialPost

ACS

Responsibilities
- knows its indentifier
- Given a user_id can find all the posts for by user
- knows its title
- knows its content
- knows any attachments

Collaborators:

Parent Class: User (User is a subclass of Model)

Child Classes: DebatePost, SocialPost

Follows

Responsibilities
- knows its indentifier
- Given a user_id can find all the posts for by user
- knows its title
- knows its content
- knows any attachments

Collaborators:

Parent Class: User (User is a subclass of Model)

Child Classes: DebatePost, SocialPost

Agrees

Responsibilities
- knows its indentifier
- Given a user_id can find all the posts for by user
- knows its title
- knows its content
- knows any attachments

Likes

Responsibilities
- knows its indentifier
- Given a user_id can find all the posts for by user
- knows its title
- knows its content
- knows any attachments

Collaborators:

Parent Class: User (User is a subclass of Model)

Child Classes: DebatePost, SocialPost

Collaborators:

Parent Class: User (User is a subclass of Model)

Child Classes: DebatePost, SocialPost

QuestionnaireResponse

Responsibilities
- knows its indentifier
- Given a user_id can find all the posts for by user
- knows its title
- knows its content
- knows any attachments

Collaborators:

Parent Class: User (User is a subclass of Model)

Child Classes: DebatePost, SocialPost

Sport

Responsibilities
- knows its indentifier
- Given a user_id can find all the posts for by user
- knows its title
- knows its content
- knows any attachments

Collaborators:

Parent Class: User (User is a subclass of Model)

Child Classes: DebatePost, SocialPost

Highlights

Responsibilities
- knows its indentifier
- Given a user_id can find all the posts for by user
- knows its title
- knows its content
- knows any attachments

Collaborators:

Parent Class: User (User is a subclass of Model)

Child Classes: DebatePost, SocialPost

Team

Responsibilities
- knows its indentifier
- Given a user_id can find all the posts for by user
- knows its title
- knows its content
- knows any attachments

Collaborators:

Parent Class: User (User is a subclass of Model)

Child Classes: DebatePost, SocialPost

Player

Responsibilities
- knows its indentifier
- Given a user_id can find all the posts for by user
- knows its title
- knows its content
- knows any attachments

Collaborators:

Parent Class: User (User is a subclass of Model)

Child Classes: DebatePost, SocialPost

Prediction

Responsibilities
- knows its indentifier
- Given a user_id can find all the posts for by user
- knows its title
- knows its content
- knows any attachments

Collaborators:

Parent Class: User (User is a subclass of Model)

Child Classes: DebatePost, SocialPost

PredictionChoice

Responsibilities
- knows its indentifier
- Given a user_id can find all the posts for by user
- knows its title
- knows its content
- knows any attachments

Collaborators:

Parent Class: User (User is a subclass of Model)

Child Classes: DebatePost, SocialPost