

Class Name: TownServiceClients	
State: baseUrl, axios	
Responsibilities	Collaborators
Manages the formats of requests and responses of different actions relates to town, including joining listing, updating, deleting, adding admins, blocking users, etc.	towns (routers)
	all implementations of TownServiceClients

Class Name: UserServiceClients	
State: baseUrl, axios	
Responsibilities	Collaborators
Manages the formats of requests and responses of users singing up, singing in, and updating user profile.	towns (routers)
	all implementations of UserServiceClients

Class Name: Town	
State: townid, password, townname, admins, creator, blcokers, isPublic, maxOccupancy	
Responsibilities	Collaborators
This is a data access object model mapping to MongoDB 'room' collection. It takes care of the actions of creating, getting room ids, deleting, and updating room information in MongoDB.	MongoDB database
	all implementations of Town

Class Name: User	
State: userName, password, email, gender, age, city	
Responsibilities	Collaborators
This is a data access object model mapping to MongoDB 'user' collection. It takes care of the actions offinding user by attributes, update user information, deleting user, and counting number of users in MongoDB.	MongoDB database
	all implementations of User

Class Name: Home	
State:	
Responsibilities	Collaborators
Being the home page component that will be displayed when the APP is started.	SignIn
	SignUp

Class Name: SignIn	
State: userName, password	
Responsibilities	Collaborators
The sign in component allows not only users to sign in with their username and password, but also keeping track of their states.	UserServiceClient
	all implementations of SignIn
	hooks that monitors the states

Class Name: SignUp	
State: userName, password	
Responsibilities	Collaborators
The sign up component allows users to create an account and keeping the states of account attributes, including username, password, email, gender, age and city.	UserServiceClient
	all implementations of SignUp
	hooks that monitors the states

Class Name: Login	
State:	
Responsibilities	Collaborators
The Login component allows user to login with user name and password.	TownServiceClient (TownJoinResponse)

Class Name: Profile	
State: userName, password, passwordToMatch, email, gender, age, city	
Responsibilities	Collaborators
The profile component allows users to update their accounts, including password, email, gender, age and city. It also renders up-to-date user information.	UserServiceClient
	all implementations of Profile
	hooks that monitors the states

Class TownSetting	
State: apiClient, currentTownID, friendlyName, isPubliclyListed, townUpdatePassword	
Responsibilities	Collaborators
This components presents information of a town at the page of game. Player is able to change town password, publicity, and friendly name of town.	all implementations of TownSetting
	hooks that monitors the states

Class TownSelection	
State: userName, newTownName, newTownIsPublic, townIDToJoin, currentPublicTowns, apiClient	
Responsibilities	Collaborators
It renders the page that includes components for users to join and create towns. It also lists all the public towns with occupancy.	all implementations of TownSelection
	hooks that monitors the states
	Profile
	TownServiceClient

Class Creator	
State: players, apiClient, curentTownID, currentTownBlockers, currentTownCreator, currentAdmininstrators, room, name	
Responsibilities	Collaborators
If the player creates a town, he/she is eligible to be a Creator. It provides the ability of blocking a user from entering the current town, assigning a new administrator etc.	all implementations of Creator
	hooks that monitors the states

Class Player	
State: location, id, name	
Responsibilities	Collaborators
Records the location, userName and id of the player.	UserLocation

Class Name: CoveyTownListener	
State:	
Responsibilities	Collaborators
Listens to player's events and get called when a player joins a town, moves, or disconnects.	Player
	CoveyTownController
	all implementations of CoveyTownListener

Class Name: CoveyTownController	
State: players, sessions, videoClients, listeners, instance	
Responsibilities	Collaborators
Stores list of players, listeners and sessions for the town. Control and manages the events that occurs in the town.	Player
	PlayerSession
	IVideoClient
	CoveyTownListener
	UserLocation
	TwilioVideo

Class Name: townJoinHandler	
State:	
Responsibilities	Collaborators
The handler manages join requests from client and returns messages if joined successfully.	Player
	PlayerSession
	CoveyTownController
	TownJoinRequest
	TownJoinResponse

Class Name: townSubscriptionHandler	
State:	
Responsibilities	Collaborators
Client uses the token returned by handler to make subscriptions to town.	Socket
	PlayerSession
	CoveyTownController
	townSocketAdapter (function)
	CoveyTownListener
	UserLocation

Class Name: IVideoClient	
State:	
Responsibilities	Collaborators
Given two strings representing town and client, issue a strong token that could authorize a client connecting to the town	all implementations of IVideoClient
	Promises
	CoveyTownController

Class Name: townAddBlockerHandler	
State:	
Responsibilities	Collaborators
This handler process a creator or admin request to add a player to block list of a town.	all implementations of townAddBlockerHandler
	CoveyTownStore
	Room (MongoDB)
	Promises
	CoveyTownController

Class Name: townAddAdminHandler	
State:	
Responsibilities	Collaborators
This handler enables the creator of town to add a user as an admin of town.	all implementations of IVideoClient
	Promises
	CoveyTownController
	CoveyTownStore
	Room (MongoDB)

Class Name: townListHandler	
State:	
Responsibilities	Collaborators
This handler manages the process of listing all the townss.	all implementations of townListHandler
	Promises
	CoveyTownStore

Class Name: singleTownListHandler	
State:	
Responsibilities	Collaborators
This handler takes in the name of town and return admins list, blockers lsit, and creator name of town.	all implementations of singleTownListHandler
	Promises
	Room (MongoDB)

Class Name: townCreateHandler	
State:	
Responsibilities	Collaborators
This handler helps to create towns in CoveyTownStore and Room model in MongoDB.	all implementations of townCreateHandler
	Promises
	CoveyTownStore
	Room (MongoDB)

Class Name: townDeleteHandler	
State:	
Responsibilities	Collaborators
This handler helps to delete towns in CoveyTownStore and Room model in MongoDB.	all implementations of townDeleteHandler
	Promises
	CoveyTownStore
	Room (MongoDB)

Class Name: townBlockerDeleteHandler	
State:	
Responsibilities	Collaborators
This handler process a creator or admin request to remove a player from block list of a town.	all implementations of townBlockerDeleteHandler
	Promises
	Room (MongoDB)

Class Name: townBlockerAdminHandler	
State:	
Responsibilities	Collaborators
This handler process a creator request to remove a player from admin list of a town.	all implementations of townBlockerDeleteHandler
	Promises
	Room (MongoDB)

Class Name: townUpdateHandler	
State:	
Responsibilities	Collaborators
This handler helps update town information.	all implementations of townBlockerDeleteHandler
	Promises
	CoveyTownStore

Class Name: checkUserByNameHandler	
State:	
Responsibilities	Collaborators
This handler checks whethere a username exists in database.	all implementations of checkUserByNameHandler
	Promises
	User (MongoDB)

Class Name: checkUserByNameAndPasswordHandler	
State:	
Responsibilities	Collaborators
This handler checks whether a user with a specified pair of username and password exists	all implementations of checkUserByNameAndPasswordHandler
	Promises
	User (MongoDB)

Class Name: createUserHandler	
State:	
Responsibilities	Collaborators
When a user sign up, this handler helps adding user profile in database.	all implementations of createUserHandler
	Promises
	User (MongoDB)

Class Name: updateUserHandler	
State:	
Responsibilities	Collaborators
This handler helps update a user account in databse.	all implementations of updateUserHandler
	Promises
	User (MongoDB)