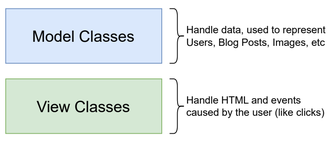
Build a Web Framework!

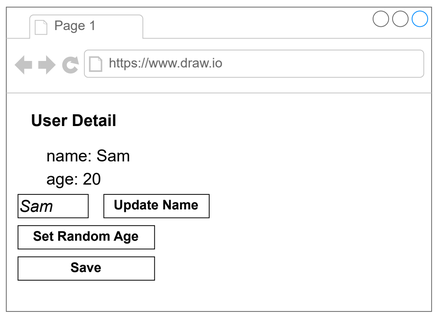

Model Classes
Handle data, used to represent
Users, Blog Posts, Images, etc
View Classes
Handle HTML and events
caused by the user (like clicks)

Probably need to create a class to represent a User and all of its data (like name and age)

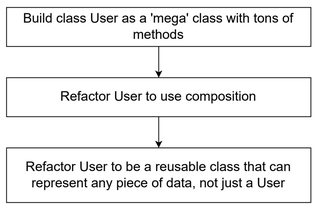
User class needs to have the ability to store some data, retrieve it, and change it

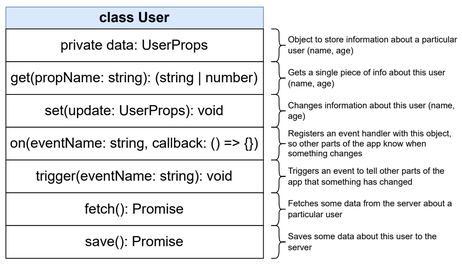
Also needs to have the ability to notify the rest of the app when some data is changed

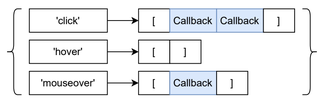
User needs to be able to persist data to an outside server, and then retrieve it at some future point

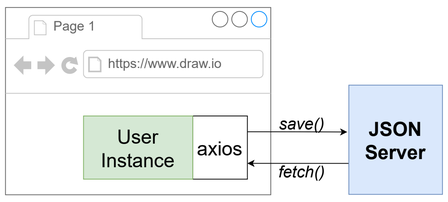


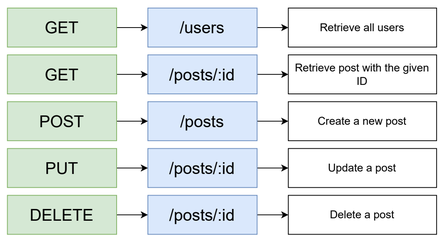
Extraction Approach




class User
private data: UserProps
get(propName: string): (string | number)
set(update: UserProps): void
on(eventName: string, callback: () => {})
trigger(eventName: string): void
fetch(): Promise
save(): Promise
Object to store information about a particular user (name, age)
Gets a single piece of info about this user (name, age)
Changes information about this user (name, age)
Registers an event handler with this object, so other parts of the app know when something changes
Triggers an event to tell other parts of the app that something has changed
Fetches some data from the server about a particular user
Saves some data about this user to the server


'click'
[
Callback
Callback
]
'hover'
[
]
'mouseover'
[
Callback
]


Page 1
https://www.draw.io
User Instance
axios
save()
fetch()
JSON Server


GET
/users
Retrieve all users
GET
/posts/:id
Retrieve post with the given ID
POST
/posts
Create a new post
PUT
/posts/:id
Update a post
DELETE
/posts/:id
Delete a post

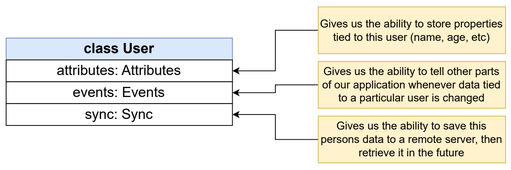
All of our models that need to be synced with a server need an 'ID' property

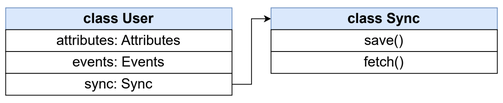
User	
prop	value
data	{ id: 1, age: 20 }

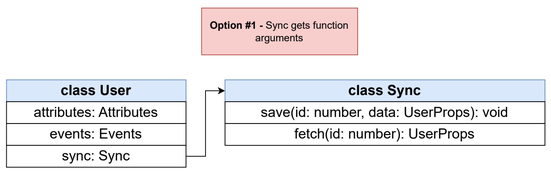
Has a server-side representation

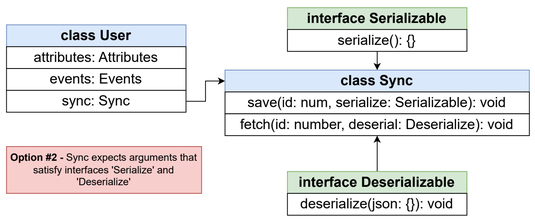
User	
prop	value
data	{ age: 20 }

Brand new! Not persisted server-side


class User
attributes: Attributes
events: Events
sync: Sync
Gives us the ability to store properties tied to this user (name, age, etc)
Gives us the ability to tell other parts of our application whenever data tied to a particular user is changed
Gives us the ability to save this persons data to a remote server, then retrieve it in the future


class User
attributes: Attributes
events: Events
sync: Sync
class Sync
save()
fetch()


Option #1 - Sync gets function arguments
class User
attributes: Attributes
events: Events
sync: Sync
class Sync
save(id: number, data: UserProps): void
fetch(id: number): UserProps


class User
attributes: Attributes
events: Events
sync: Sync
interface Serializable
serialize(): {}
class Sync
save(id: num, serialize: Serializable): void
fetch(id: number, deserial: Deserialize): void
interface Deserializable
deserialize(json: {}): void
Option #2 - Sync expects arguments that satisfy interfaces 'Serialize' and 'Deserialize'

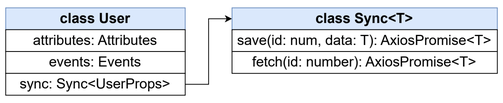
Serialize

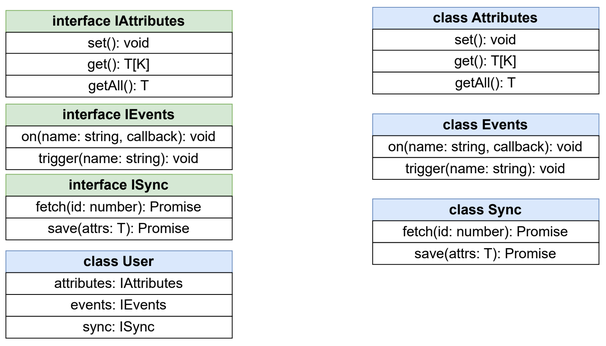
Convert data from an object into some save-able format (json)

Deserialize

Put data on an object using some previously saved data (json)

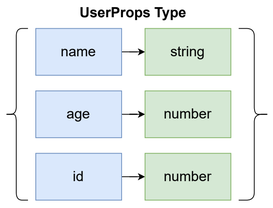
Option #3 - Sync is a generic class to customize the type of 'data' coming into save()

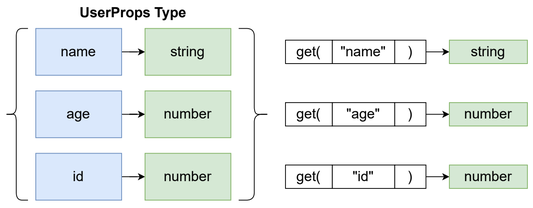



interface IAttributes
set(): void
get(): T[K]
getAll(): T
interface IEvents
on(name: string, callback): void
trigger(name: string): void
interface ISync
fetch(id: number): Promise
save(attrs: T): Promise
class User
attributes: IAttributes
events: IEvents
sync: ISync

class Attributes
set(): void
get(): T[K]
getAll(): T
class Events
on(name: string, callback): void
trigger(name: string): void
class Sync
fetch(id: number): Promise
save(attrs: T): Promise

1	In Typescript, strings can be types
2	In JS (and therefore TS), all object keys are strings


UserProps Type
name
string
age
number
id
number


UserProps Type
name
string
age
number
id
number
get(
"name"
)
string
get(
"age"
)
number
get(
"id"
)
number