

## Part 2: Creating New Stories

In this part, you'll design and write the functionality to let logged-in users add new stories. We've broken this task into two parts. It will help you to tackle them in this order.

### Subpart 2A: Sending Story Data to the Backend API

Here, you'll need to write a method that adds a new story by sending the right data to our API.

We've given you a comment string and a stub method for this, ***addStory***, in the ***StoryList*** class. Complete this function, making sure your function takes in the same parameters and returns the same result as our comment said.

Test that this works, and that your method returns an instance of ***Story***. You can do this in the browser console with:

```
let newStory = await storyList.addStory(currentUser,
  {title: "Test", author: "Me", url: "<http://meow.com>"});
```

And make sure that returns an instance of the ***Story*** class:

```
newStory instanceof Story;    // should be true!
```

### Subpart 2B: Building The UI for New Story Form/Add New Story

Now, we'll add the UI for the story-adding feature:

- Add a form in the HTML for the story. This should initially be hidden.
- Add a link in the navbar with the text of "submit".
- Write a function in ***nav.js*** that is called when users click that navbar link. Look at the other function names in that file that do similar things and pick something descriptive and similar.
- Write a function in ***stories.js*** that is called when users submit the form. Pick a good name for it. This function should get the data from the form, call the ***.addStory*** method you wrote, and then put that new story on the page.

## Part 3: Favorite stories

In this step, you'll add a feature marking/unmarking a story as a favorite.

As before, it's best to write the data-logic and API-call part first, and do the UI afterwards.

## Subpart 3A: Data/API Changes

Allow logged in users to “favorite” and “un-favorite” a story. These stories should remain favorited when the page refreshes.

Allow logged in users to see a separate list of favorited stories.

**The methods for adding and removing favorite status on a story should be defined in the User class.**

## Part 4: Removing Stories

Allow logged in users to remove a story. Once a story has been deleted, remove it from the DOM and let the API know its been deleted.

## Further Study

- Add some error handling for when a username has already been taken or if credentials are incorrect!
- Allow users to edit stories they have created.
- Add a section for a “user profile” where a user can change their ***name*** and ***password*** in their profile.
- Style the application so that it is presentable on mobile devices.
- Add infinite scroll! When a user scrolls to the bottom of the page, load more stories.
- Come up with some other features you can build using what our Hack or Snooze API makes available to you!