# Exercise: Source Control Systems

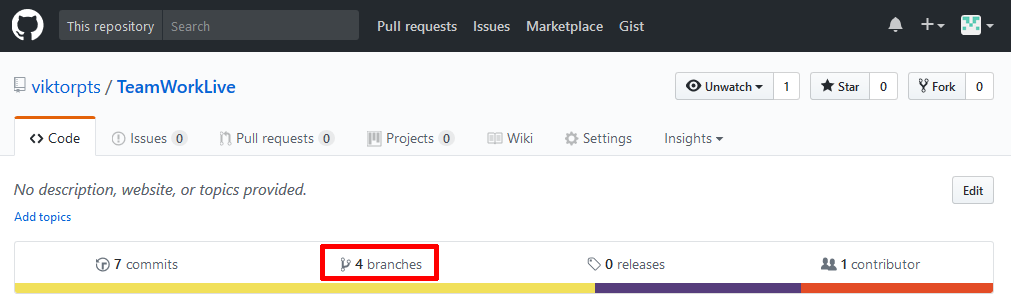
# Stage 7, Task 3 – Merge Detailed-View into master

**To begin, make sure Task 7-2 is finished and committed to the master.**

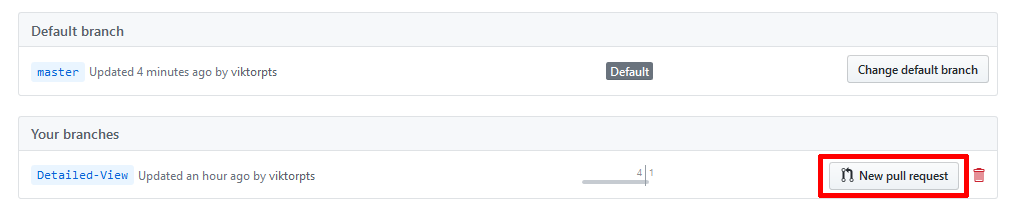
It’s likely the three features we developed in branches may conflict with each other – it’s time to pull them into the master one by one and resolve any merge conflicts that have been created.

We’re going to create a **pull request** for branch Detailed-View. A pull request signals the readiness of our branch to be merged into the production level of development (or into an upstream branch, that may later go into production). Then the manager of the project may review the changes and approve our request for code to be pulled from our branch into the master. For larger or sensitive projects, it’s a good idea to give the right to approve pulls into the master only to senior level developers, or just the project lead.

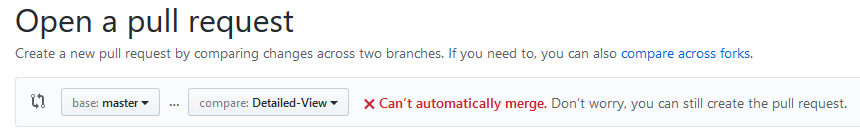
Open your project’s **GitHub page** and navigate to the **“branches”** section:



Click **“New pull request”** beside the branch **“Detailed-View”**:

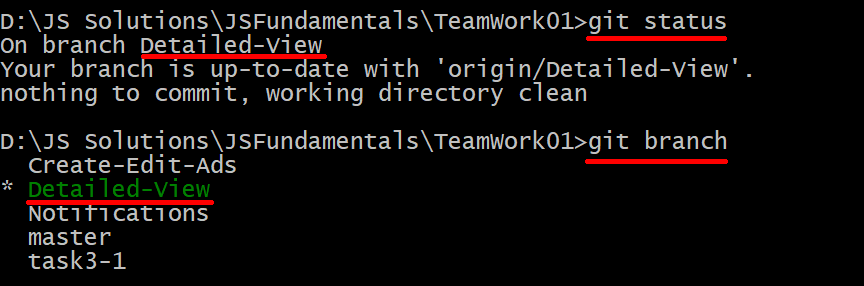


At this point, it’s likely you’ll see this message:



You can still create the pull request, but resolving conflicts from the GitHub page is not very elegant. What you need to do now is **pull the master into your branch locally**, using your favorite method (bash, GUI, IDE plugin).

To pull from the master trough the console, first make sure you’re in the correct local branch by typing git status or git branch. You should be in branch Detailed-View:



Pull from the remote master into the current branch by typing:

git fetch origin

git merge origin/master

This will bring your local master up-to-date with the remote and then merge the changes into the currently active branch (Detailed-View). It will also mark any conflicts inside the files. Note that if you use WebStorm, a resolution window will pop-up automatically. If you merged the changes from the console, but wish to resolve the conflicts inside the IDE, click **VCS** 🡪 **Git** 🡪 **Resolve conflicts…**, otherwise you’ll have to manually edit the files in your favorite editor.

The properties that we added to the adverts aren’t applied to either of the create or edit views and their respective properties. Due to code similarity in the back-end script, the loading of single adverts has been completely mangled with the create and edit endpoint functions. In **WebStorm’s conflict resolver**, you’ll see a much cleaner version of the changes to be merged, so it **is recommended** you do that. If not, follow the instructions bellow.

First of all, it might be easier to compare the two branches before the merge attempt and manually transfer the changes. If you chose to go that route, **back out of the merge** by typing (note the double dash):

git reset --hard

The **easiest thing** to do is probably to keep your branch open and clone the master in another folder, open both files and copy/paste the changes **manually**.

|  |
| --- |
| Bellow you will find **only the additions** to functions that are required to bring the create/edit/load requests and endpoints up to speed with the **new properties** of ads (description and image). |

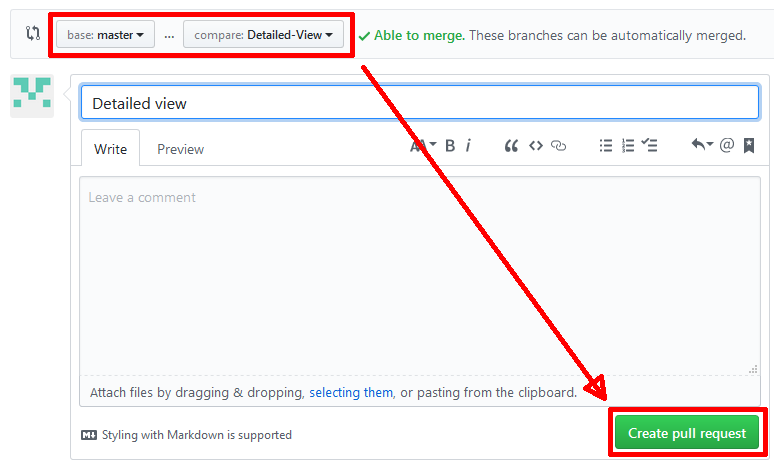
|  |
| --- |
| index.html |
| *↑*  *Existing code here …*  <**section id="viewCreateAd" class="viewCreateAd"**>  <**h1 class="titleForm"**>Create new Advertisement</**h1**>  <**form id="formCreateAd" class="form"**>  <**div**>Title:</**div**>  <**div**><**input type="text" name="title" required** /></**div**>  <**div**>Description:</**div**>  <**div**><**textarea name="description" rows="10" required**></**textarea**></**div**>  <**div**>Date Published:</**div**>  <**div**><**input type="date" name="datePublished" required** /></**div**>  <**div**>Price:</**div**>  <**div**><**input type="number" step="0.01" name="price" required** /></**div**>  <**div**>Image:</**div**>  <**div**><**input type="text" name="image" required** /></**div**>  <**div**><**input type="button" id="buttonCreateAd" value="Create"** /></**div**>  </**form**> </**section**>  <**section id="viewDetailsAd" class="viewDetailsAd"**> </**section**>  <**section id="viewEditAd" class="viewEditAd"**>  <**h1 class="titleForm"**>Edit existing advertisement</**h1**>  <**form id="formEditAd" class="form"**>  <**div**><**input type="hidden" name="id" required** /></**div**>  <**div**><**input type="hidden" name="publisher" required** /></**div**>  <**div**>Title:</**div**>  <**div**><**input type="text" name="title" required** /></**div**>  <**div**>Description:</**div**>  <**div**><**textarea name="description" rows="10" required**></**textarea**></**div**>  <**div**>Date Published:</**div**>  <**div**><**input type="date" name="datePublished" required** /></**div**>  <**div**>Price:</**div**>  <**div**><**input type="number" step="0.01" name="price" required** /></**div**>  <**div**>Image:</**div**>  <**div**><**input type="text" name="image" required** /></**div**>  <**div**><**input type="button" id="buttonEditAd" value="Edit"** /></**div**>  </**form**> </**section**>  *Existing code here …*  *↓* |

|  |
| --- |
| prodavachnik.js |
| *↑*  *Existing code here …*  **function** afterPublisherRequest(publisher) {  **let** advertData = {  title: $(**'#formCreateAd input[name=title]'**).val(),  description: $(**'#formCreateAd textarea[name=description]'**).val(),  publisher: publisher.username,  datePublished: $(**'#formCreateAd input[name=datePublished]'**).val(),  price: Number($(**'#formCreateAd input[name=price]'**).val()),  image: $(**'#formCreateAd input[name=image]'**).val()  };   **const** kinveyAdvertsUrl = kinveyBaseUrl + **"appdata/"** + kinveyAppKey + **"/adverts"**;  $.ajax({  method: **"POST"**,  url: kinveyAdvertsUrl,  headers: kinveyAuthHeaders,  data: advertData,  success: createAdvertSuccess,  error: handleAjaxError  });   **function** createAdvertSuccess(response) {  listAdverts();  showInfo(**'Advertisement created.'**);  } }  *↑*  *Existing code here …*  *↓*  **function** loadAdvertForEditSuccess(advert) {  $(**'#formEditAd input[name=id]'**).val(advert.\_id);  $(**'#formEditAd input[name=title]'**).val(advert.title);  $(**'#formEditAd input[name=publisher]'**).val(advert.publisher);  $(**'#formEditAd textarea[name=description]'**).val(advert.description);  $(**'#formEditAd input[name=datePublished]'**).val(advert.datePublished);  $(**'#formEditAd input[name=price]'**).val(advert.price);  $(**'#formEditAd input[name=image]'**).val(advert.image);  showView(**'viewEditAd'**); }  *↑*  *Existing code here …*  *↓*  **function** displayAdvert(advertId){  **const** kinveyAdvertUrl = kinveyBaseUrl + **"appdata/"** +  kinveyAppKey + **"/adverts/"** + advertId;  **const** kinveyAuthHeaders = {  **'Authorization'**: **"Kinvey "** + sessionStorage.getItem(**'authToken'**),  };   $.ajax({  method: **"GET"**,  url: kinveyAdvertUrl,  headers: kinveyAuthHeaders,  success: displayAdvertSuccess,  error: handleAjaxError  });   $(**'#viewDetailsAd'**).empty();   **function** displayAdvertSuccess(advert) {  **let** advertInfo = $(**'<div>'**).append(  $(**'<img>'**).attr(**"src"**, advert.image),  $(**'<br>'**),  $(**'<label>'**).text(**'Title:'**),  $(**'<h1>'**).text(advert.title),  $(**'<label>'**).text(**'Description:'**),  $(**'<p>'**).text(advert.description),  $(**'<label>'**).text(**'Publisher:'**),  $(**'<div>'**).text(advert.publisher),  $(**'<label>'**).text(**'Date:'**),  $(**'<div>'**).text(advert.datePublished));   $(**'#viewDetailsAd'**).append(advertInfo);   showView(**'viewDetailsAd'**);  }  }   *// advertisement/edit POST* **function** editAdvert() {  **const** kinveyAdvertUrl = kinveyBaseUrl + **"appdata/"** + kinveyAppKey +  **"/adverts/"** + $(**'#formEditAd input[name=id]'**).val();  **const** kinveyAuthHeaders = {  **'Authorization'**: **"Kinvey "** + sessionStorage.getItem(**'authToken'**),  };   **let** advertData = {  title: $(**'#formEditAd input[name=title]'**).val(),  description: $(**'#formEditAd textarea[name=description]'**).val(),  publisher: $(**'#formEditAd input[name=publisher]'**).val(),  datePublished: $(**'#formEditAd input[name=datePublished]'**).val(),  price: $(**'#formEditAd input[name=price]'**).val(),  image: $(**'#formEditAd input[name=image]'**).val()  };   $.ajax({  method: **"PUT"**,  url: kinveyAdvertUrl,  headers: kinveyAuthHeaders,  data: advertData,  success: editAdvertSuccess,  error: handleAjaxError  });   **function** editAdvertSuccess(response) {  listAdverts();  showInfo(**'Advertisement edited.'**);  }  } } |

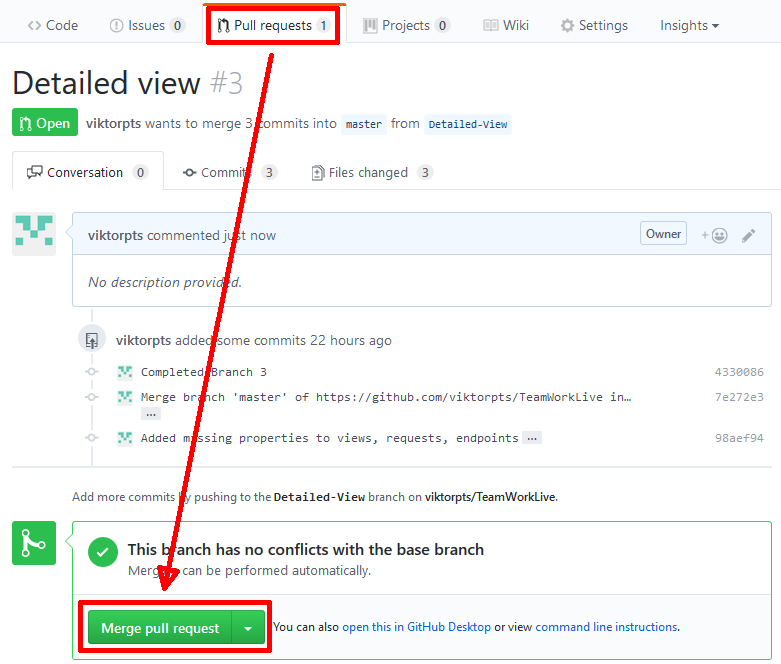
|  |
| --- |
| api.js |
| *↑*  *Existing code here …*  *↓*  *// Create advert* $.mockjax(**function** (requestSettings) {  **if** (requestSettings.url === **"https://mock.api.com/appdata/kid\_rk/adverts"** &&  requestSettings.method === **"POST"**) {  **return** {  response: **function** (origSettings) {  **if** (requestSettings.headers[**"Authorization"**].includes(**"Kinvey mock\_token"**)) {  **let** data = requestSettings.data;  **let** lastId = 0;  **if** (adverts.length > 0) {  lastId = adverts.map(a => a.\_id).sort((a, b) => b - a)[0];  }  **let** token = requestSettings.headers[**"Authorization"**].replace(**"Kinvey "**, **""**);  **let** creator = users.filter(u => u.\_kmd.authtoken === token)[0].\_id;  **let** advert = {  \_id: ++lastId,  \_acl: {  creator: creator  },  title: data.title,  description: data.description,  publisher: data.publisher,  datePublished: data.datePublished,  price: data.price  };  adverts.push(advert);  **this**.responseText = advert;  } **else** {  **this**.status = 403;  **this**.responseText = **"You are not authorized"**;  }  }  };  } });  *↑*  *Existing code here …*  *↓*  *// Edit advert* $.mockjax(**function** (requestSettings) {  **if** (requestSettings.url.match(/https:\/\/mock\.api\.com\/appdata\/kid\_rk\/adverts\/(.+)/) &&  requestSettings.method === **"PUT"**) {  **let** advertId = Number(requestSettings.url.match(/https:\/\/mock\.api\.com\/appdata\/kid\_rk\/adverts\/(.+)/)[1]);  **return** {  response: **function** (origSettings) {  **if** (requestSettings.headers[**"Authorization"**].includes(**"Kinvey mock\_token"**)) {  **let** advert = adverts.filter(a => a.\_id === advertId);  **let** data = requestSettings.data;  **if** (advert.length > 0) {  advert = advert[0];  advert.title = data.title;  advert.description = data.description;  advert.publisher = data.publisher;  advert.datePublished = data.datePublished;  advert.price = data.price;  advert.image = data.image;  **this**.responseText = advert;  }  **this**.responseText = {};  } **else** {  **this**.status = 403;  **this**.responseText = **"You are not authorized"**;  }  }  };  }  }); })(); |

|  |
| --- |
| **If you can’t make sense of the code or your best effort still couldn’t un-mangle the changes, do not be afraid to copy all files from folder 7. Final and commit them to the branch Detailed-View. After that you can create a pull request and merge into the master normally.** |

Once everything is cleaned up, we can create a pull request, as instructed above. There **should be no more conflicts** and the green message “**Able to merge**” will appear. Leave a note if you wish and make sure the target is set to “master”:



After the request is created, the page where it can be managed should open automatically. If not, you can see it from “Pull requests” in the navigation pane. If you see the green button “Merge pull request”, then congratulations, you can seamlessly merge the branch into the master without conflicts!



Make sure you pull the master locally and test the application before proceeding with the next task!

|  |
| --- |
| **This is the final task. Congratulate yourself and your teammates and submit a link to your repository for evaluation on the** [course page](https://softuni.bg/trainings/1686/practical-teamwork-javascript-may-2017) **under Exercises: Git and GitHub. Well done!** |