ToDo Application (Open Classrooms Project)

Introduction

To-do-app is an application which helps to manage tasks. The user can add new task to to-do list, update and easy delete it, separately or whole list when the tasks are completed. The app has minimalistic design and simply functionality.

The application has been created by using Model-View-Controller(MVC) – an architectural design pattern that separates three main functionalities: Model – View – Controller. MVC has been used here to add dynamical functionality: user can add new task without reloading the webpage (like Single Page Application – SPA).

Model as a central component manages data logic and methods. Here are created prototypes to manage a local storage object and main app functionality like adding, updating and deleting task of the list.

View as an output representor displays and manages the user interactions with the application, manipulates DOM structure and represents its functionality.

Controller as a third part of the MVC pattern connects model and view by converting inputs from the View for the Model component.

Detailed description of all functions.

- Controller Object– controls interactions between Model and View.
  - Parameters: model object and view object.
  - Prototypes:
    - setView (loads and initialize the view),
    - showAll (displays all items in the todo-list),
    - showActive (renders uncompleted tasks),
    - showComplited(renders completed tasks),
    - addItem (creates new todo task, saving it in the local storage by adding ID),
    - editItem (starts editing mode of todo task by matching with the correct ID),
    - editItemSave (successfully edits item and save the changing by using matched ID),
    - editItemCancel (cancels the item editing mode),
    - removeItem (removes item from to-do-list and storage by using its ID as a parameter),
    - removeCompletedItems (removes all completed tasks),
    - toggleComplete (gives ID and updates the state of completeness of task in the storage),
    - toggleAll (change the state of completeness of the tasks: on/off),

- Model Object - creates new Model instance and connects it with the storage.
  - Parameters: storage object.
  - Prototypes:
    - create (creates a new todo model and saves it in the storage),
    - read (finds and returns a model in storage, if the query isn't given, returns everything),
    - update (updates a model, every action based on unique ID),
    - remove (removes a model from storage),

        - removeAll (removes all data from storage),
        - getCount (counting active, completed and total tasks by finding the in the storage).
- **Storage Object** – manages data storage by using the local session storage.
- **Helpers** - a bunch of helper methods for querying the selectors and encapsulating the DOM.
- **Template** – delivers template function to display list items, change button states, escape characters.
- **View Object** – manipulates DOM structures attached to user interaction. It has two simple entry points:
    - bind (takes a todo application event and registers the handler),
    - render (renders the given command with the options).

Bugs Fixed

- Fixed spelling error at line no. 95
    - Location js/controller.js
- Missing Id in input tag at line no.16
    - Location index.html
- Added Date.now in line no. 90 instead of function to assign id.
    - Location js/store.js
- Removed unnecessary code(function to print console in production version) at line no. 165
    - Location js/controller.js

Automatic Jasmine unit testing.

In this Jasmine dubbing process was required to add some tests to already written ones. New tests have to check following cases:

- 'should show entries on start-up'
    - the 'todo' array should be empty, when the application starts;
- 'should show all entries without "all" route'
    - shows total count of the tasks, array can be empty or filled it with the tasks;
- 'should show active entries'
    - the completed tasks which are set to false (completed = false);
- 'should show completed entries'
    - the completed tasks which are set to true (completed = true);
- 'should show the content block when todos exists'
    - create a list of the tasks, when they exist;
- 'should highlight "All" filter by default'
    - sets 'all' as default, takes total count, even if it's empty;
- 'should toggle all todos to completed'
    - updates all tasks as completed (model component);
- 'should update the view'
    - updates the status as completed (view component);
- 'should add a new todo to the model'
    - adds new task to the list;
- 'should remove an entry from the model'
    - removes todo task (model component), empty array.

Audit Performance

Audit to-do-app

The audit of to-do app was performed using Developer Tools in Microsoft Edge browser on windows machine.

Results:

- page loads fast, because it based only on html, css and vanilla.js technologies,
- page doesn't need a large amount of memory, because it doesn't require any media files,
- application is simple, without any heavy fonts, animations or complicated styles to be load.

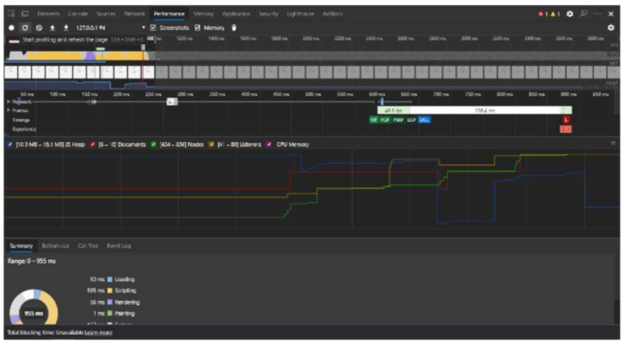
Comparison

	Todo-app	Todo-net
Loading Time	381ms	5.96s
transferred	44.3kB	2.3MB

Performance

Todo-app

Performance Report



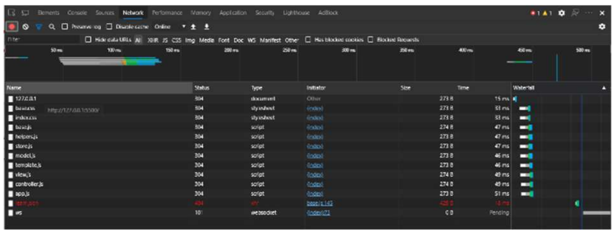
Range: 0 – 955 ms

50 ms	Loading
595 ms	Scripting
56 ms	Rendering
1 ms	Painting

955 ms

Total blocking time: Unavailable Learn more

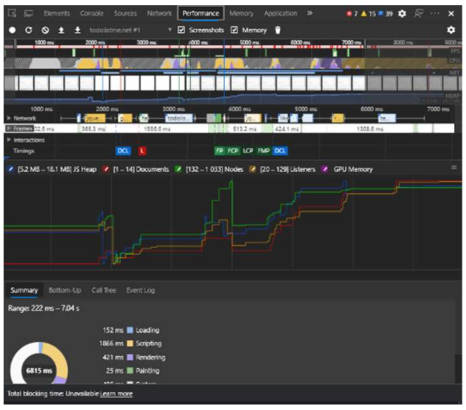
Network Report



Name	Status	Type	Initiator	Size	Time
127.0.0.1	304	document	Other	273 B	15 ms
base.css	304	stylesheet	(index)	273 B	33 ms
index.css	304	stylesheet	(index)	273 B	33 ms
base.js	304	script	(index)	274 B	47 ms
helpers.js	304	script	(index)	273 B	47 ms
store.js	304	script	(index)	273 B	47 ms
model.js	304	script	(index)	273 B	46 ms
template.js	304	script	(index)	273 B	46 ms
view.js	304	script	(index)	274 B	49 ms
controller.js	304	script	(index)	274 B	49 ms
app.js	304	script	(index)	273 B	51 ms
[illegible]	404	[illegible]	base.js:143	[illegible]	[illegible]
ws	101	websocket	(index):73	0 B	Pending

Todo-net

Performance Report



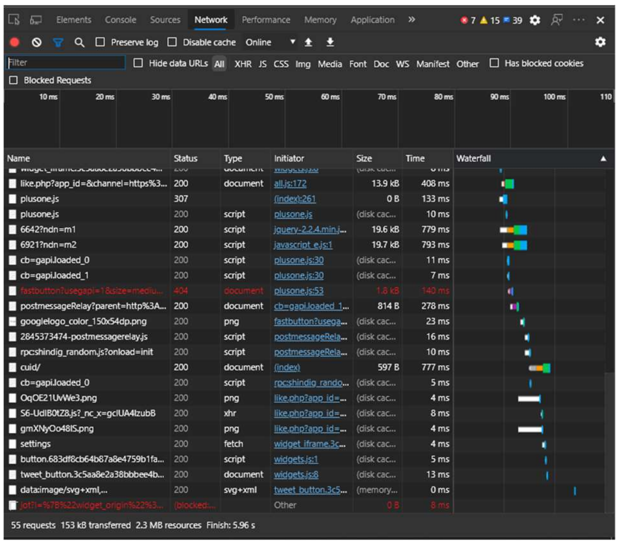
Range: 222 ms – 7.04 s

152 ms	🟦 Loading
1866 ms	🟨 Scripting
421 ms	🟪 Rendering
25 ms	🟩 Painting

6815 ms

Total blocking time: Unavailable Learn more

Network Report



Name	Status	Type	Initiator	Size	Time
like.php?app_id=&channel=https%3...	200	document	all.js:172	13.9 kB	408 ms
plusone.js	307		(index):261	0 B	133 ms
plusone.js	200	script	plusone.js	(disk cac...	10 ms
6642?ndn=m1	200	script	jquery-2.2.4.min.j...	19.6 kB	779 ms
6921?ndn=m2	200	script	javascript_e.js:1	19.7 kB	793 ms
cb=gapi.loaded_0	200	script	plusone.js:30	(disk cac...	11 ms
cb=gapi.loaded_1	200	script	plusone.js:30	(disk cac...	7 ms
fastbutton?usegapi=1&size=mediu...	404	document	plusone.js:53	1.8 kB	140 ms
postmessageRelay?parent=http%3A...	200	document	cb=gapi.loaded_1...	814 B	278 ms
googlelogo_color_150x54dp.png	200	png	fastbutton?usega...	(disk cac...	23 ms
2845373474-postmessagerelay.js	200	script	postmessageRela...	(disk cac...	16 ms
rpc:shindig_random.js?onload=init	200	script	postmessageRela...	(disk cac...	10 ms
cuid/	200	document	(index)	597 B	777 ms
cb=gapi.loaded_0	200	script	rpc:shindig_rando...	(disk cac...	5 ms
OqOE21UvWe3.png	200	png	like.php?app_id=...	(disk cac...	4 ms
S6-UdIB0tZ8.js?_nc_x=gclUA4lzubB	200	xhr	like.php?app_id=...	(disk cac...	8 ms
gmXNyOo48IS.png	200	png	like.php?app_id=...	(disk cac...	4 ms
settings	200	fetch	widget_iframe.3c...	(disk cac...	4 ms
button.683df8cb64b87a8e4759b1fa...	200	script	widgets.js:1	(disk cac...	5 ms
tweet_button.3c5aa8e2a38bbbee4b...	200	document	widgets.js:8	(disk cac...	13 ms
data:image/svg+xml,...	200	svg+xml	tweet_button.3c5...	(memory...	0 ms
jot?l=%7B%22widget_origin%22%3...	(blocked:...		Other	0 B	8 ms

55 requests 153 kB transferred 2.3 MB resources Finish: 5.96 s

Advantage of Todo-app

- Simple design
- Basic functionality, easy to understand for a user
- Based on MVC model, which is easy to read and develop
- Low data transfer
- Low memory consumption

Disadvantage of Todo-app

- Only local storage in use, not possible to save the data for longer period

Advantage of Todo-net

- The data can be saved locally and remotely, after registration

Disadvantage of Todo-net

- Very slow loading time
- Memory consumption
- Google Ads delay loading and increases data transfer
- Image used is not optimized as background image is 132kB