

Lisp in Summer Projects Submission

Submission Date	2013-10-24 13:41:45
Full Name	Peter Barnett
Country	USA
Project Name	Todo-Repl
Type of software	web app
General category	web tool
LISP dialect	Clojure
GitHub URL	https://github.com/Pance/todo-repl-webapp
Did you start this project?	Yes, all the code is written by me
Project Description	I want to describe my project in this form.
Purpose	TODO-REPL provides users with a traditional TODO list that they can manipulate with Clojure in a REPL prompt. TODO-REPL will also allow users to customize the behavior of their todo-list by writing Clojure code. Examples of behavior that could be customized would be repeating tasks, sorted/filtered todo lists or creating new tasks.
Function	TODO-REPL provides users with a Clojure REPL prompt that users can write code into to interact with their todo-list. In TODO-REPLs current state, users can add new tasks in the form of a Clojure map data structure and mark specific tasks as complete.
Motivation	I was tired of the rigid nature of my current todo-list application. I wanted a todo-list application that would perfectly fit my workflows and could adapt in behavior as my workflows changed.
Audience	The target audience of TODO-REPL is hackers who keep todo lists as well as users of todo-lists who want a more flexible and customizable solution.

Methodology

An individual TODO is represented as a Clojure map which can contain a number of keys that represent the different attributes of a TODO, such as the name of the task, a due date, completion status, task context, tag labels and etc. A user's todo-list is a vector of maps. Hiccup was used in a function to build the HTML for a single task based on a given task (the task-item function in ns `todo_repl_webapp.views`). This function was mapped over the user's list of tasks to generate HTML for a list of tasks. The web application consists of a single page with AJAX used to send expressions to the server for evaluation and to display the results on the page.

Conclusion

I learned a lot about writing Clojure code but also designing web applications in the development of this project. I completed all the features I had planned on implementing for this version of the project.

Future features for TODO-REPL are:
User authentication and user todo-lists
More task attributes (notes, URLs, priorities, etc.)
Sorted Lists
Filtered Lists
REPL textbox input prediction
Responsive views for mobile use
An API
Replace JS on frontend with ClojureScript
Integrate with third party applications
- Email, so that tasks can be created by email
- Pocket, so that items saved to pocket will show in todo-list

Build Instructions

With leiningen installed and the project cloned with git:

```
$ cd todo-repl-webapp  
$ lein deps  
$ lein compile
```

Though these commands are optional. You can run the execution command without running the above commands.

Test Instructions

There are no tests :(

Execution Instructions

With leiningen installed and inside the project directory:

```
$ lein run 8000
```

Describe any bugs or caveats

There are only a limited number of functions that work correctly when entered in the REPL prompt on the webapp (the ones shown in the instructions). A more complete Clojure REPL was intended to be provided.

There is only one single todo-list maintained by the application. So multiple users using this application at the same time will modify the same todo-list.

Screen shots

□ [pbarnett_todo-repl_1.jpg](#)

todo-repl

Instructions

To Display Tasks
(tasks)
Tasks look like this
{:name name
:due date (try natural language)
:context context
}

To Reset Tasks
(init-tasks)
To Add a Task
(add-new-task
{:name "Build a Rocketship"
:due "tomorrow"
:context "hobbies"})

To Complete a Task
(complete-task index)

```
(add-new-task {:name "Submit my LISP project" :due "now" :context "programming"})
```

Eval

list

0	Go get some milk	shopping	Fri Oct 25 16:41:35 UTC 2013
1	Go get new shoes	shopping	Fri Oct 25 16:41:35 UTC 2013
2	Clean the kitchen sink	cleaning	never

[pbarnett_todo-repl_2.jpg](#)

todo-repl

Instructions

To Display Tasks
(tasks)
Tasks look like this
{:name name
:due date (try natural language)
:context context
}

To Reset Tasks
(init-tasks)
To Add a Task
(add-new-task
{:name "Build a Rocketship"
:due "tomorrow"
:context "hobbies"})

To Complete a Task
(complete-task index)

Eval

list

0	Go get some milk	shopping	Fri Oct 25 16:41:35 UTC 2013
1	Go get new shoes	shopping	Fri Oct 25 16:41:35 UTC 2013
2	Clean the kitchen sink	cleaning	never
3	Submit my LISP project	programming	Thu Oct 24 16:45:33 UTC 2013

[pbarnett_todo-repl_3.jpg](#)

todo-repl

Instructions

To Display Tasks

```
(tasks)
```

Tasks look like this

```
{:name name  
 :due date (try natural language)  
 :context context  
 }
```

To Reset Tasks

```
(init-tasks)
```

To Add a Task

```
(add-new-task  
 {:name "Build a Rocketship"  
  :due "tomorrow"  
  :context "hobbies"})
```

To Complete a Task

```
(complete-task index)
```

(complete-task 1)

Eval

list

0	Go get some milk	shopping	Fri Oct 25 16:41:35 UTC 2013
1	Go get new shoes	shopping	Fri Oct 25 16:41:35 UTC 2013
2	Clean the kitchen sink	cleaning	never
3	Submit my LISP project	programming	Thu Oct 24 16:45:33 UTC 2013

[pbarnett_todo-repl_4.jpg](#)

todo-repl

Instructions

To Display Tasks

```
(tasks)
```

Tasks look like this

```
{:name name  
 :due date (try natural language)  
 :context context  
 }
```

To Reset Tasks

```
(init-tasks)
```

To Add a Task

```
(add-new-task  
 {:name "Build a Rocketship"  
  :due "tomorrow"  
  :context "hobbies"})
```

To Complete a Task

```
(complete-task index)
```

Eval

list

0	Go get some milk	shopping	Fri Oct 25 16:41:35 UTC 2013
1	Go get new shoes	shopping	Fri Oct 25 16:41:35 UTC 2013
2	Clean the kitchen sink	cleaning	never
3	Submit my LISP project	programming	Thu Oct 24 16:45:33 UTC 2013

[pbarnett_todo-repl_5.jpg](#)

todo-repl

Instructions

To Display Tasks

(tasks)
Tasks look like this

```
{:name name
 :due date (try natural Language)
 :context context
 }
```

To Reset Tasks

(init-tasks)
To Add a Task

```
(add-new-task
 {:name "Build a Rocketship"
 :due "tomorrow"
 :context "hobbies"})
```

To Complete a Task

```
(complete-task index)
```

```
(complete-task 3)
```

Eval

list

0	Go get some milk	shopping	Fri Oct 25 16:41:35 UTC 2013
1	Go get new shoes	shopping	Fri Oct 25 16:41:35 UTC 2013
2	Clean the kitchen sink	cleaning	never
3	Submit my LISP project	programming	Thu Oct 24 16:45:33 UTC 2013

[pbarnett_todo-repl_6.jpg](#)

todo-repl

Instructions

To Display Tasks

(tasks)
Tasks look like this

```
{:name name
 :due date (try natural Language)
 :context context
 }
```

To Reset Tasks

(init-tasks)
To Add a Task

```
(add-new-task
 {:name "Build a Rocketship"
 :due "tomorrow"
 :context "hobbies"})
```

To Complete a Task

```
(complete-task index)
```

Eval

list

0	Go get some milk	shopping	Fri Oct 25 16:41:35 UTC 2013
1	Go get new shoes	shopping	Fri Oct 25 16:41:35 UTC 2013
2	Clean the kitchen sink	cleaning	never
3	Submit my LISP project	programming	Thu Oct 24 16:45:33 UTC 2013

[pbarnett_todo-repl_7.jpg](#)

Official

I have read rules and have abided by them.
I am 18 years of age or older.
I am not living in Brazil, Quebec, Saudi Arabia, Cuba, Iran,
Myanmar (Burma), North Korea, Sudan, or Syria.

