

Proveta - JS patterns & testing

Spam Filter

Il tuo team sta sviluppando un Spam filter per una azienda. Dopo un primo incontro vi siete messo d'accordo in consegnare quanto spiegato in seguito.

Mail filter



Before the filter

Mail filter



After the filter

Primo sprint:

1. All'apertura della pagina all' utente viene visualizzato il elenco di messaggi sul database.
2. All' cliccare su "filter" il vostro spam engine dovrebbe restituire i messaggi che non sono spam.

In questa fase, l'azienda vi ha fornito un database di messaggi e regole di spam in questo formato:

```
var rules = ["spam.com", "compratutto.it"];  
var msgsg = ["news@spam.com", "carlo@gmail.com", "jessy@compratutto.it"];
```

Il vostro team ha deciso di strutturare la gestione dei filtri - per facilitare il lavoro degli sviluppatori e tester - in questo modo:

```
var MailModel = {  
  Initialises the model with the "database" of filter rules  
  and messages  
  init : function(){  
  },  
  
  Filters out messages in the "database" that match  
  the spam rules. Returns an array of messages,  
  excluding those that match the filter rules.  
  filter : function(){  
  }  
};
```

Example of usage:

```
MailModel.init();  
MailModel.filter()  
-> ["carlo@gmail.com", "trentose2@googlegroups.com"]
```

Your task:

- 1- Implement Spam Filter (we suggest using js patterns)
- 2- Implement Test Cases for MailModel.

Proveta - JS patterns & testing

Guidelines

- Fork the repository at <https://github.com/mbaezpy/trentose-mailfilter.git>
- Create a branch sprint1, and commit all your changes here

Directory structure:

- index.html -> il documento principale
- app.js -> qui dovete scrivere la logica della vostra app
- data.js -> qui si trovano i dati di prova. Non modificare questo file.
- test
 - SpecRunner.html -> to run the tests
 - spec/app.spec.js -> here you should write your test cases

Secondo Sprint

At some point during the exam we will announce a change in the requirements.