Team Hovercraft



**Idea Summary**

**What is the Code Games Idea?**

The idea is that users often need to look at details of entities in varius system, if we provide the most frequently used details in a simpel tooltip this will prevent the user from having to drill down as the required information is directly available, this would cause less pages to be requested from our servers. This in return reduces server load.

The idea was always to implement this in a universal way so it could be used on any website that could benefit from a few tooltips here and there.

Once we started with this many new use cases arrised, we can use them to give more information about certain buttons, we can use them for onboarding in new features

**Advantages:**

* Works on any website (for which tooltips are configured)
* Prevent users from unnecessary drill downs by providing useful contextual help
* Reduces server load by lower page requests
* It can help with a detailed and interactive walkthroughs / new feature announcements / user onboarding​

**How it is seposed to works:**

An application sends the tooltips it wants to be available to our backend, those tooltips are stored on a per website (url) locator (element to attach it to) and context (such as logged in user / company / organisation).

An application also keep the tooltips up to date incase data inside the application changes that requires the tooltip to be updated.

When a client then load a webpage, it requests what elements are avaiable for tooltips given the url / context, and the server then send a list of locators.

When the client then hovers over an element that marked to receive a tooltip, the tooltip is requested from the server on the fly.

The client side is seposed to be a javascript libaray that’s included in the applications its websites.

**How the prototype works:**

For speed and efficiency we cut out the application inserting and updating part, out tooltips in the prototype are hardcoded.

When a client loads a webpage all tooltips and their locators are loaded at the start.

When the client then hovers over an element the tooltip is created on the fly and remove when the mouse leaves the element.

The client side for the prototype is a chrome extension as that allows us to inject our script to websites we don’t own for demonstration purposes.

**How the setup the prototype:**

To get the tooltips of the prototype working, you need to install the chrome extension:

In chrome options, goto “more tools” -> “Extensions”.

At the top right enabled developer mode.

Press the “Load Unpackaged” button.

Select the “chromeExtension” folder of our source code.

Reload any webpage atleast once after the extension is added.

**Websites with tooltips:**

<https://hovercraft.azurewebsites.net/> - Timespinner bosses list, this is Hovercraft main test website, its included in the server side code

<https://ospokolenko.wixsite.com/nonsense> - Our 2nd testing website

<https://login.twinfield.com/> - For Twinfield credentials are required, (this availability of the account below is limited)

User: Hovercraft  
 Password: Hovercraft01

Organisation: TESTPROD\_AZURE

**Source Code structure**

Describe purpose of each folder/project

|  |  |
| --- | --- |
| Hovercraft\chromeExtension | The chrome extension that had to be installed by clients to see the tooltips |
| Hovercraft\Site\Hovercraft\Hovercraft.sln | The backend solution file |
| Hovercraft\Site\Hovercraft\Hovercraft | The backend code base, holds both the dummy site aswel as the tooltip api |

**What tools technologies were used to create?**

|  |  |
| --- | --- |
| Documents | Existing docx document |
| SourceControl | GitHub |
| API | .Net 4.7 Web.Api |
| FrontEnd | Jquery, CSS, JS |
| Hosting platfom | Microsoft Azure |
| BackEnd Language | C#, JSON |

**Deployement of backend:**

User supplied code, Build it, if any errors occur you might need to install .Net Framework 4.7.

When its build you can publish it.

The application published to Microsoft Azure using normal web publishment, it was created using Microsoft’s template for web.api applications, and can be published to your local iis if needed, Once its published, you should see the timespinner bosses page, and the api will be available under /api/tooltips.

Note: if you want to run the backend locally you change the endpoint url inside the chrome extenion

In file: Hovercraft\chromeExtension\content.js, line 3. After making any changes to the chrome extension be sure to relaod the extension and the webpages