

Obtendo placar

Transcrição

Agora que nosso placar já está sendo salvo no servidor, seria interessante que ele fosse carregado automaticamente assim que a página fosse aberta. Vamos começar a implementar isto criando a função `atualizaPlacar()` :

```
//placar.js

function atualizaPlacar(){

}
```

Esta função será a responsável por fazer um GET para a url que contém o placar, trazer os dados, e adicioná-los ao HTML assim que a página for carregada.

Buscando dados do placar

Vamos começar pelo get:

```
//placar.js

function atualizaPlacar(){
  $.get("http://localhost:3000/placar",function(){

  });
}
```

Queremos receber os dados do placar, então precisamos receber a variável *data* como argumento da função `$.get()`, pois ela que irá conter os dados que buscamos na URL:

```
//placar.js
function atualizaPlacar(){
  $.get("http://localhost:3000/placar",function(data){

  });
}
```

Iterando pelos dados recebidos

Sabemos que na *data* recebemos o array que contém o placar, então para iterar por ele vamos utilizar a nossa conhecida função `.each()`, para ganhar acesso aos objetos individuais que contém a pontuação de cada linha do placar:

```
function atualizaPlacar(){
  $.get("http://localhost:3000/placar",function(data){
    data.each(function(){
```

```
    });  
  });  
}
```

Se tentássemos executar o código acima, receberíamos um erro na cara, pois arrays do Javascript **não** tem acesso a função `.each()`, apenas objetos jQuery. Por isso, vamos envolver a variável `data` com a função `$` para transformá-la em objeto jQuery:

```
function atualizaPlacar(){  
  $.get("http://localhost:3000/placar",function(data){  
    $(data).each(function(){  
  
      });  
    });  
}
```

Bom, se temos acesso a cada objeto dentro do array `data`, podemos construir linhas da tabela para inserir estes dados no HTML. Para a criar a linha, podemos utilizar a função `novaLinha(usuario,palavras)`, que havíamos criado em um dos capítulos anteriores. Esta função já é capaz de receber o nome do usuário e o número de palavras e criar um linha HTML para ser adicionada ao placar, basta passarmos os dados do objeto que estamos iterando:

```
function atualizaPlacar(){  
  $.get("http://localhost:3000/placar",function(data){  
    $(data).each(function(){  
      var linha = novaLinha(this.usuario, this.pontos);  
    });  
  });  
}
```

Com a linha criada, podemos adicioná-la no HTML, dentro do `<tbody>`, com a função `.append()`:

```
function atualizaPlacar(){  
  $.get("http://localhost:3000/placar",function(data){  
    $(data).each(function(){  
      var linha = novaLinha(this.usuario, this.pontos);  
      $("tbody").append(linha);  
    });  
  });  
}
```

Carregando o placar junto da página

Fica faltando apenas chamar a função `atualizaPlacar()` assim que a página é carregada, dentro de `main.js`:

```
//main.js  
$(function() {  
  atualizaTamanhoFrase();  
  inicializaContadores();  
  inicializaCronometro();  
});
```

```
    inicializaMarcadores();  
    $("#botao-reiniciar").click(reiniciaJogo);  
  
    atualizaPlacar();  
  });
```

Pronto! Agora o placar deve vir preenchido corretamente assim que a página for carregada!

O que aprendemos:

- Como carregar o placar assim que a página abrir
- Relembrando as funções: \$.get, .each(), .append(), \$(function(){})