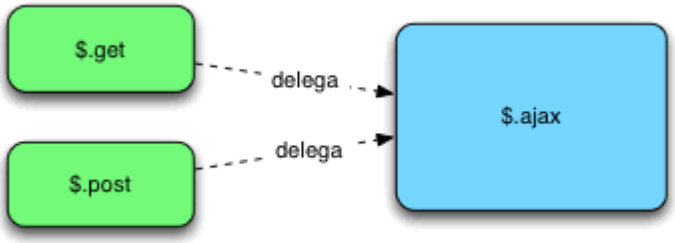
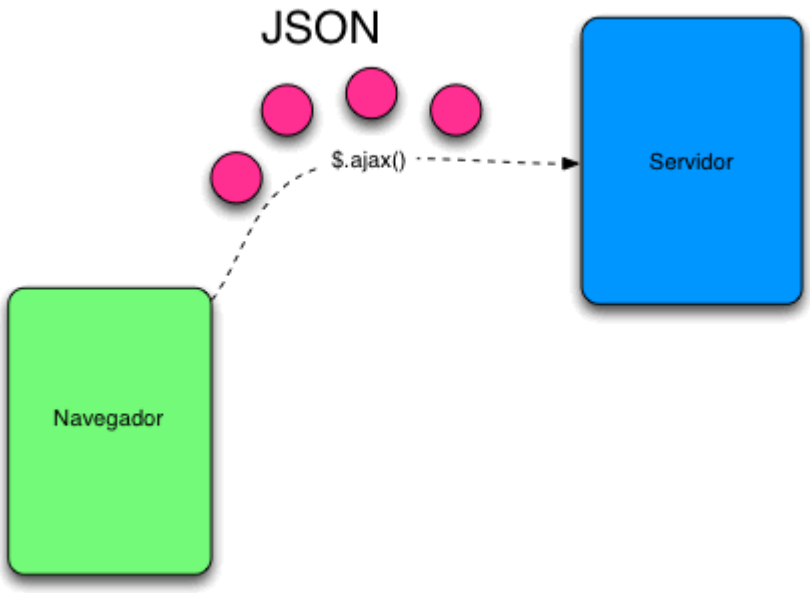
Estamos muy acostumbrados a trabajar con jQuery y sus capacidades de Ajax. La mayor parte de las veces nos apoyamos en los métodos $.get y $.post ya que nos permiten realizar las operativas más habituales en el mundo Ajax (enviar información y recibirla) . Sin embargo estos dos métodos no son más que sencillas fachadas sobre sobre el método $.ajax que es el principal de jQuery a la hora de realizar este tipo de peticiones.



Usando jQuery $.ajax() y JSON

Hay situaciones en las que nos puede ser interesante seguir usando $.ajax() por ejemplo una de las más clásicas es cuando queremos enviar al servidor información en formato JSON y no de la forma clásica que se envía par clave valor. Recordemos que cada día hay mas frameworks que soportan de una forma natural la recepción de peticiones HTTP cuyos datos usen este formato.



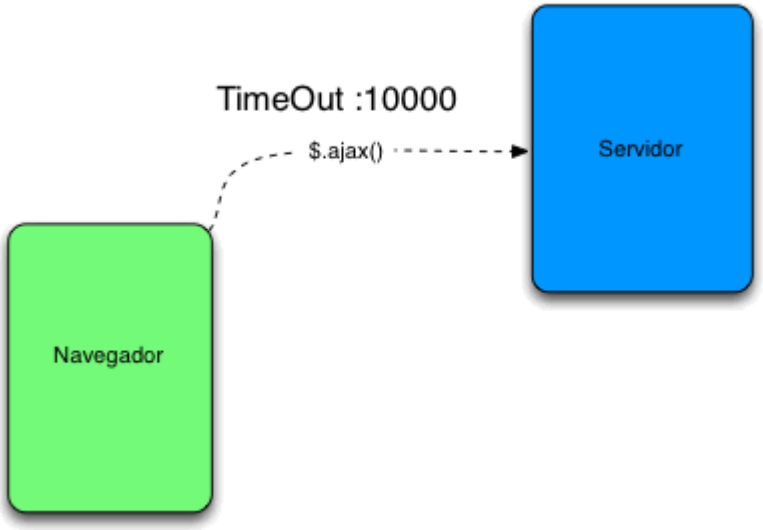
Para enviar estos datos usaremos el siguiente bloque de código:

```
$.ajax({
        type: "POST",
        url: 'facturas',
        contentType: 'application/json',
        data: JSON.stringify(factura),
        });
```

De esta forma jQuery enviará los datos en formato JSON para su posterior procesamiento.

Timeouts y $.ajax()

Otra de las capacidades interesantes que tiene $.ajax() es la posibilidad de definir un tiempo de timeout a nuestra medida. Existen situaciones en las que el timeout por defecto (de navegador) puede no ser el ideal así pues podemos configurar el nuestro a medida.



El código es muy sencillo basta con añadir el parámetro:

```
$.ajax({
        timeout:10000,
        type: "GET",
        url: 'facturas',
        });
```

Cada día es más importante conocer a detalle $.ajax() ya que las exigencias a nivel de este tipo de peticiones es cada día más alta.

Otros artículos relacionados: Servicios REST , JQuery eventos Globales ,JQuery Tuning