Considerations for Client Caches

Elton Stoneman

elton@sixeyed.com

pluralsight
hardcore developer training

Outline

- **Cache keys**
  - Request content
- **Browser caching**
  - jQuery and AJAX
  - Same-origin security policy
- **Cross-domain requests**
  - JSONP
  - CORS

Client Cache Keys

- **Protocol**
  - Proxies - HTTPS possible, but not typical
  - “Modern” browsers – HTTP and HTTPS
- **Method**
  - GET and HEAD
  - POST possible, not typical
- **URL**
  - Full, including querystrings (usually)

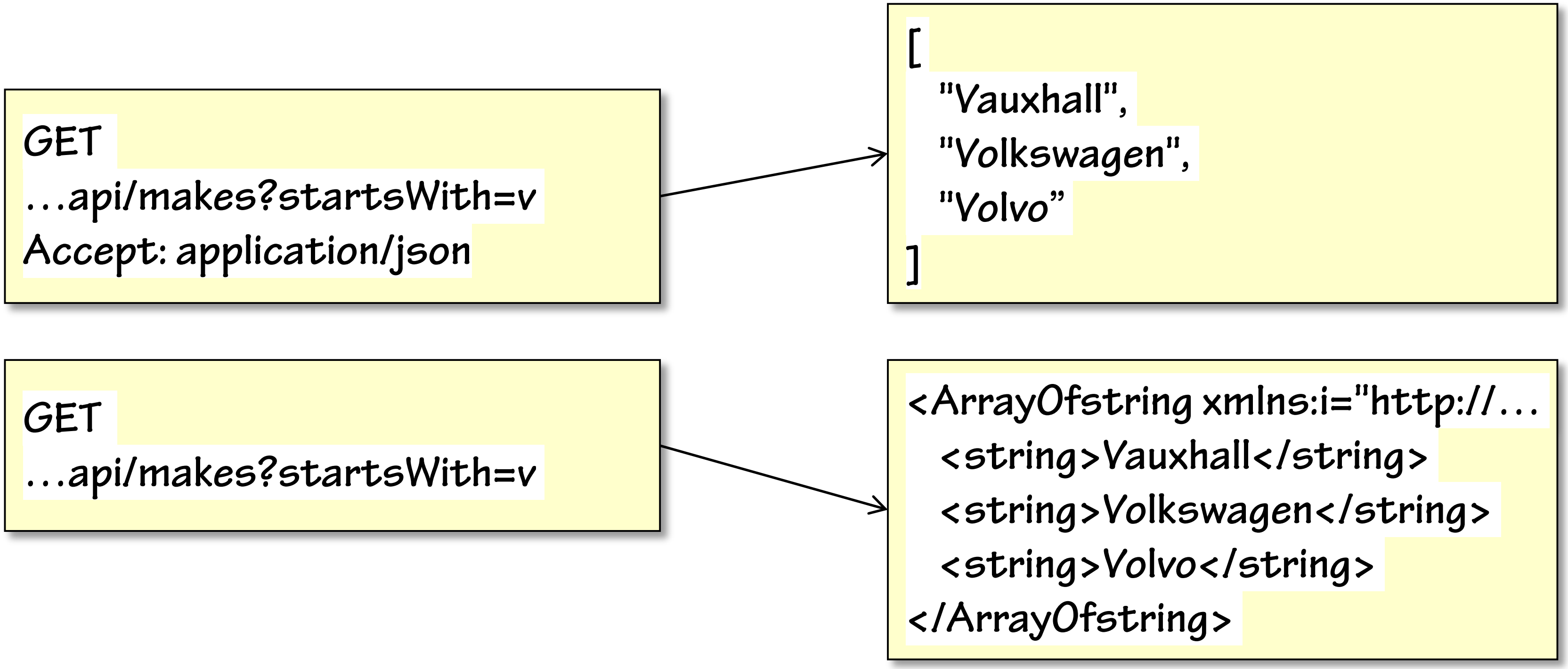
```
GET http://api.valetmymotor.eu/api/makes?startsWith=v
```

```
GET http://api.valetmymotor.eu/api/makes?startsWith=f
```

```
GET http://api.valetmymotor.eu/api/makes?startsWith=v&userId=123
```

Client Cache Keys

- **Request headers**



```
GET
…api/makes?startsWith=v
Accept: application/json
```

```
[
  "Vauxhall",
  "Volkswagen",
  "Volvo”
]
```

```
GET
…api/makes?startsWith=v
```

```
<ArrayOfstring xmlns:i="http://…
  <string>Vauxhall</string>
  <string>Volkswagen</string>
  <string>Volvo</string>
</ArrayOfstring>
```

Client Cache Keys

- Demo

Client Cache Keys

- **Vary response header**
  - Vary: Accept
  - Vary: Accept, X-Api-Version
  - Vary: *
- **Cache key**
  - hash(Method, URL, Vary[])
  - URL – replaced if varied
- **Cache variations**
  - Vary response
  - Query strings
    - …api/vehicles/1
    - …api/vehicles?id=1

jQuery and AJAX

- **Platform-dependent client cache**
  - .NET WebClient with RequestCachePolicy
- **Client call direct to API**
  - Responsive design
  - Distributed solution, bypass Web server
  - HTTP, routed through browser & proxy caches

```
var data = {
    "makeName": makeName,
    "modelName": modelName
};
$.getJSON(Vehicles.baseUrl, data, successCallback);
```

```
GET …/api/vehicles?makeName=Audi&ModelName=A3
```

jQuery and AJAX

- Demo

jQuery and AJAX

- **$.getJSON**
  - Calls routed through browser + cache
  - Cache enabled by default
- **Same-origin policy**
  - Security feature in all browsers
  - AJAX calls across origins fail silently

http://www.valetmymotor.eu/myquotes

http://www.valetmymotor.eu/api/makes?startsWith=f

http://api.valetmymotor.eu/makes?startsWith=f

Cross-domain requests

- **JSONP**
  - JSON-with-padding
  - <script> exempt from same-origin policy
- **Client specifies callback**

```
<script src="…/api/makes?startsWith=v&callback=resp”/>
```

- **Server pads response**

```
resp(["Vauxhall", "Volkswagen“,"Volvo”]);
```

- **Client executes callback**

Cross-domain requests

- **JSONP and jQuery**
  - □ Supported with $.ajax method
  - □ Specific configuration

```
$.ajax({ type: 'GET',
      url : "…/api/vehicles?makeName=Audi&ModelName=A3",
      dataType: 'jsonp',
      jsonpCallback: ‘cb’,
      cache : true…
```

- **JSONP and WebAPI/WCF**
  - □ Custom implementation

Cross-domain requests

- JSONP demo

Cross-domain requests

- **JSONP**
  - jQuery support
  - Caching needs careful configuration
- **Concerns**
  - Security – XSS & XSRF vulnerabilities
  - No error handling – works or fails
- **Wide browser support**
  - Has been a popular approach
  - Becoming deprecated

Cross-domain requests

- **CORS**
  - Cross-Origin Resource Sharing
  - Normal AJAX request
  - Content negotiation
- **Client request header**
  - “Modern” browser
  - Origin: http://www.valetmymotor.eu
- **Server response header**
  - Access-Control-Allow-Origin: http://www.valetmymotor.eu
- **CORS and WebAPI/WCF**
  - Custom implementation

Cross-domain requests

- CORS demo

Cross-domain requests

- **CORS**
  - Standards-based
  - Secure
  - Configurable
  - Normal HTTP transaction
- **Limited client support**
  - Early support in Firefox and Chrome
  - iOS and Android
  - All major browsers from 2013 versions
- **Potential caching impact**
  - Access-Control-Allow-Origin wildcard
  - Vary response

Summary

- **Cache keys**
  - URLs for maximum cache use
  - GET …api/vehicles/1
  - Content negotiation -> Vary response header
- **jQuery**
  - AJAX calls to API
  - Browser blocks cross-domain
- **JSONP**
- **CORS**