

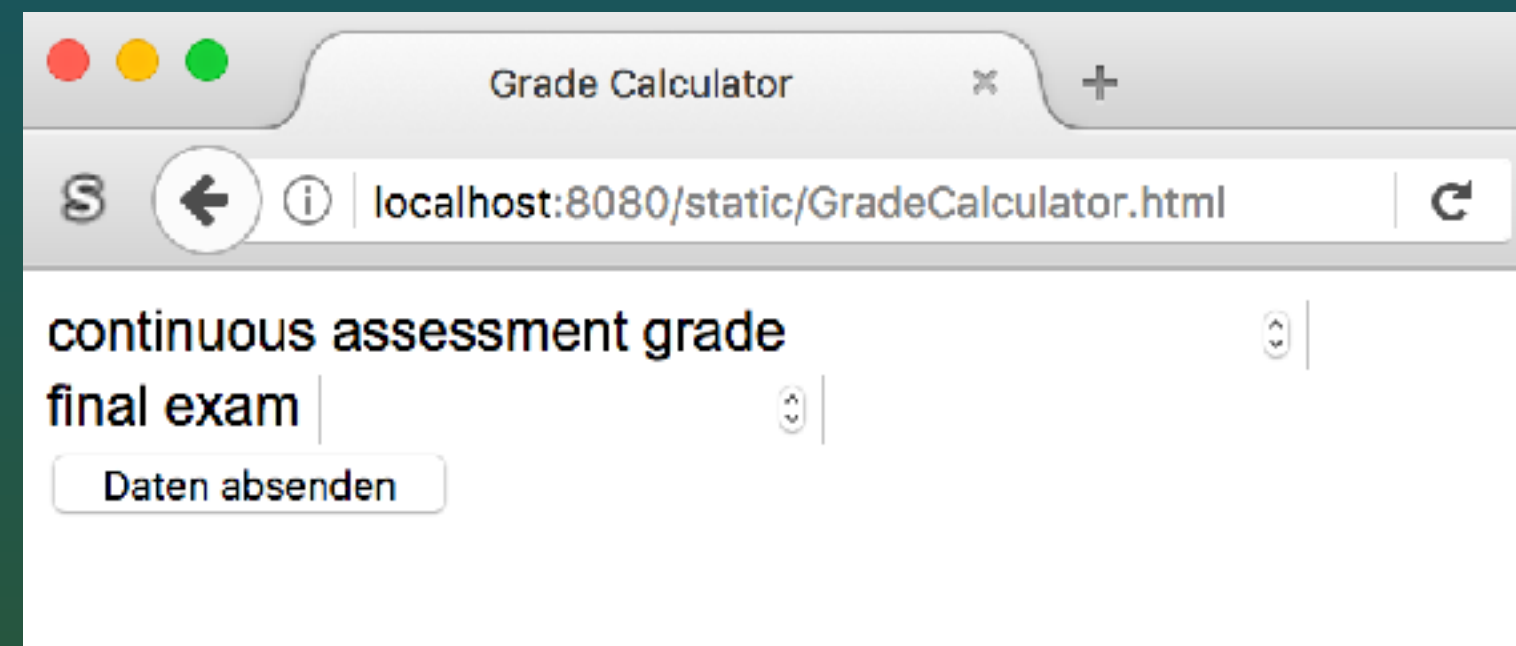
# Services

Web Engineering

Prof. D. König

# Three Tier Web Applications

*Client*



*Server*



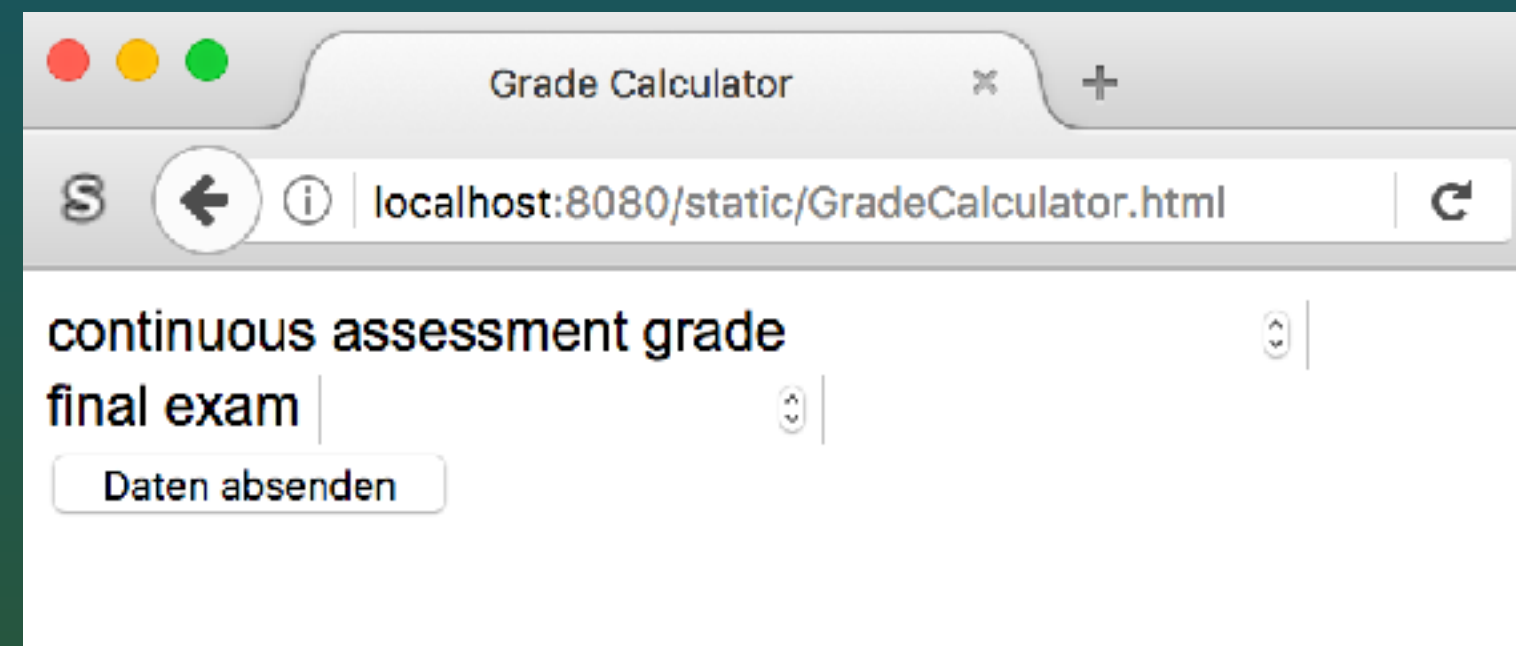
*DB*



*persist*

# Server with Service

*Client*



*Server*



*Service*



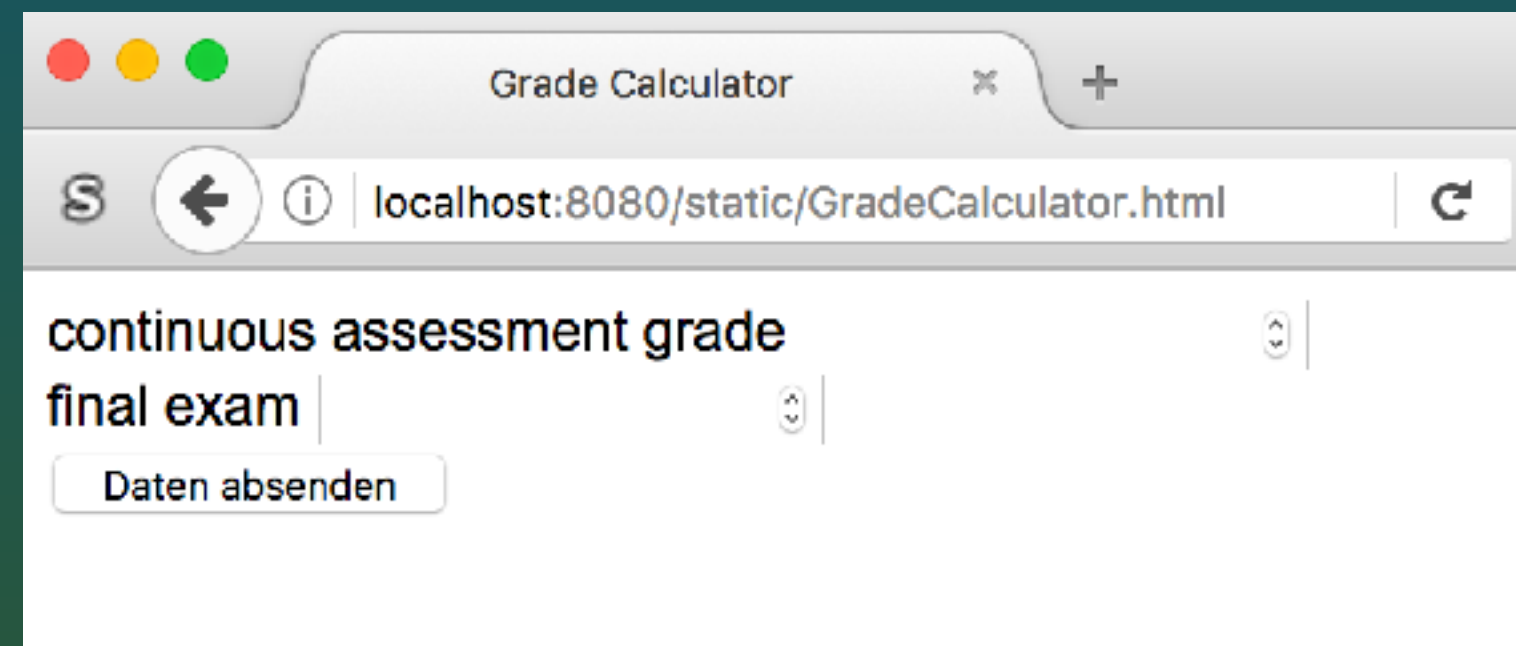
*req/resp*

# "Smart UI" with Service

*Client*

*Server*

*Service*



*req/resp*

# JavaScript Service APIs

[https://www.flickr.com/services/api/  
request.rest.html](https://www.flickr.com/services/api/request.rest.html)

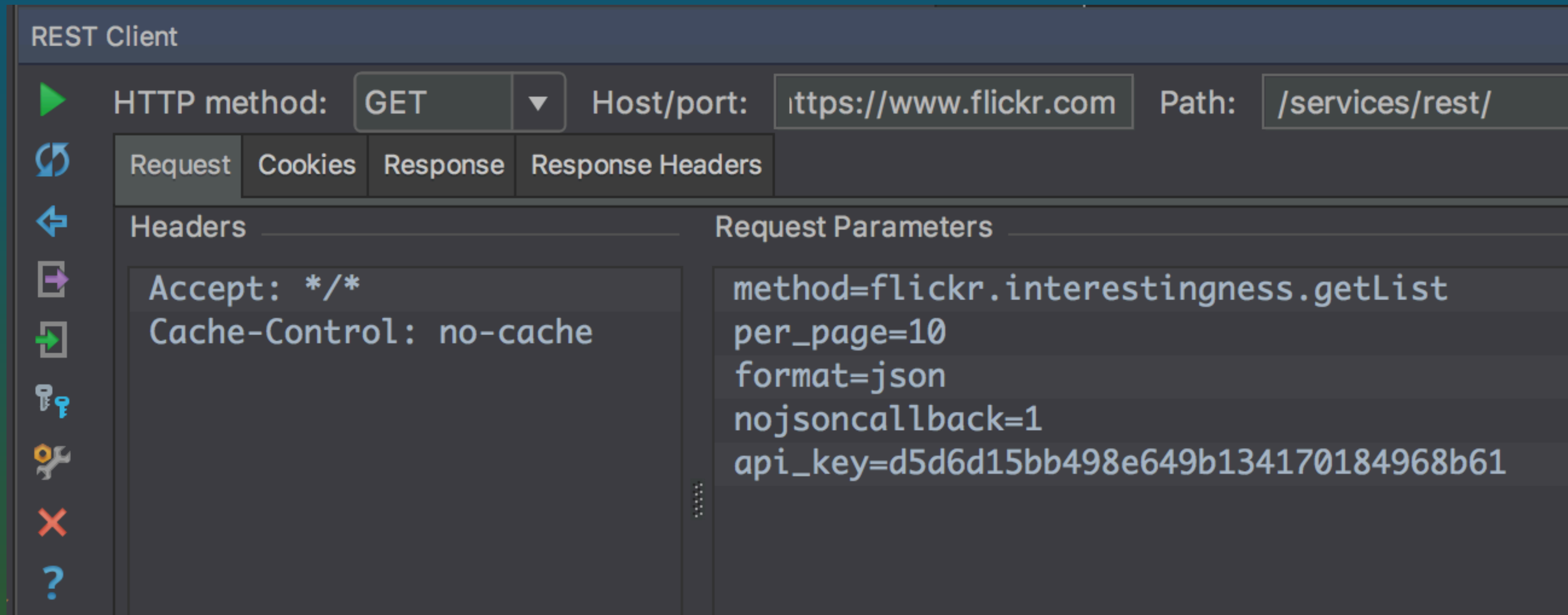
Get a developer key (use mine carefully)

Use key to request a secret

Use secret to request a resource



# Integrated REST Client



# Result

REST Client

HTTP method:  Host/port:  Path:

Request Cookies Response Response Headers

```
{
  "photos": {
    "page": 1,
    "pages": 50,
    "perpage": 10,
    "total": 500,
    "photo": [
      {
        "id": "2849",
        "server": "2849",
        "farm": 3,
        "title": "Stump Lake 4.2",
        "ispublic": 1,
        "isfriend": 0,
        "isfamily": 0,
        "secret": "be35ff66ae",
        "owner": "82828410@N02"
      },
      {
        "id": "2913",
        "server": "2913",
        "farm": 3,
        "title": "Picking up pop",
        "ispublic": 1,
        "isfriend": 0,
        "isfamily": 0,
        "secret": "b0880c5edf",
        "owner": "82828410@N02"
      },
      {
        "id": "2943",
        "server": "2943",
        "farm": 3,
        "title": "Light...Bodie Lighthouse, Outer Banks, North Carolina",
        "ispublic": 1,
        "isfriend": 0,
        "isfamily": 0,
        "secret": "74aa1570d3",
        "owner": "57866871@N03"
      },
      {
        "id": "33748943162",
        "server": "2874",
        "farm": 3,
        "title": "Light...Bodie Lighthouse, Outer Banks, North Carolina",
        "ispublic": 1,
        "isfriend": 0,
        "isfamily": 0,
        "secret": "3e488aae5e",
        "owner": "33050285@N03"
      },
      {
        "id": "33868667896",
        "server": "9969966",
        "farm": 3,
        "title": "Light...Bodie Lighthouse, Outer Banks, North Carolina",
        "ispublic": 1,
        "isfriend": 0,
        "isfamily": 0,
        "secret": "9969966",
        "owner": "9969966@N03"
      }
    ]
  }
}
```

JSON

# Result

```
{ "photos": {  
  "page": 1, "pages": 50, "perpage": 10, "total": 500,  
  "photo": [  
    {  
      "id": "33090986453",  
      "owner": "133058989@N03",  
      "secret": "5ce24df9f7",  
      "server": "2849",  
      "title": "Stump Lake 4.2"  
    }, ...  
  ]  
}
```



# Fetch Resource URL Pattern

`http://static.flickr.com/  
server-id/id_secret_m.jpg`

# URL

Uniform Resource Locator

protocol, host, port, path, query, fragment

scheme:[//[user:password@]host[:port]][/]path[?query][#fragment]

# HTTP(s)

Hyper Text  
Transfer Protocol  
Very simple  
Line/delimiter based

```
# First request
```

```
GET /quote?symb=YH00 HTTP/1.1
```

```
Host: www.example.org
```

```
HTTP/1.1 200 OK
```

```
Content-Type: text/plain; charset=UTF-8
```

```
15.96
```

```
# Second request 10 minutes later
```

```
GET /quote?symb=YH00 HTTP/1.1
```

```
Host: www.example.org
```

```
HTTP/1.1 200 OK
```

```
Content-Type: text/plain; charset=UTF-8
```

```
16.10
```

# HTTP Verbs

Verb	Change	Use
<b>GET</b>	No	Single or collective read, may be cached
<b>PUT</b>	Yes	Modify resource in place
<b>POST</b>	Yes	Create new resource
<b>DELETE</b>	Yes	Delete resource
HEAD	No	
OPTIONS	No	

# REST

## Representational State Transfer

- Resource addressing by URL
- Uniform operations by HTTP verbs
- Self-containment (no conversational state)
- Choice of format (XML, JSON, ...)



# REST

Easy to get started - easy to get lost.

Issues: type safety, versioning, docs

Alternatives:

SOAP (WSDL), CORBA, EJB, \*-RPC, RMI,  
local services

# REST in Grails

<http://docs.grails.org/latest/guide/webServices.html>

Note the usage of `@Resource` in domain classes and "respond" instead of "render" in the controller actions.