

# Intro to Load Testing your Mendix applications

Gonçalo Marcos

2018-09-27



# Agenda

- Intro to Load Testing – What, Why and How
- Load Testing – Concepts and Context
- Load Testing with Mendix
- JMeter
- Demo 1 – Create a test script using a recording tool
- Bottlenecks, Metrics, Improvements
- Demo 2 – Optimized vs Non-Optimized pages
- Analyze Results
- Demo 3 – How to create a record
- Load testing - Best practices

# Load Testing

- **What?**
  - Put application/system under load
- **Why?**
  - Real life conditions
  - Know how the application will behave when load increases
  - Identify performance issues and bottlenecks



# Load Testing

- **How?**
  - Requests/sec
    - More users
    - Same users but less time between actions
      - Increased use of application
    - More use of services (service intensive app)
  - Data (3 months vs 12 months)





# Stress Test?

- Going beyond the load test
- Find the breaking point

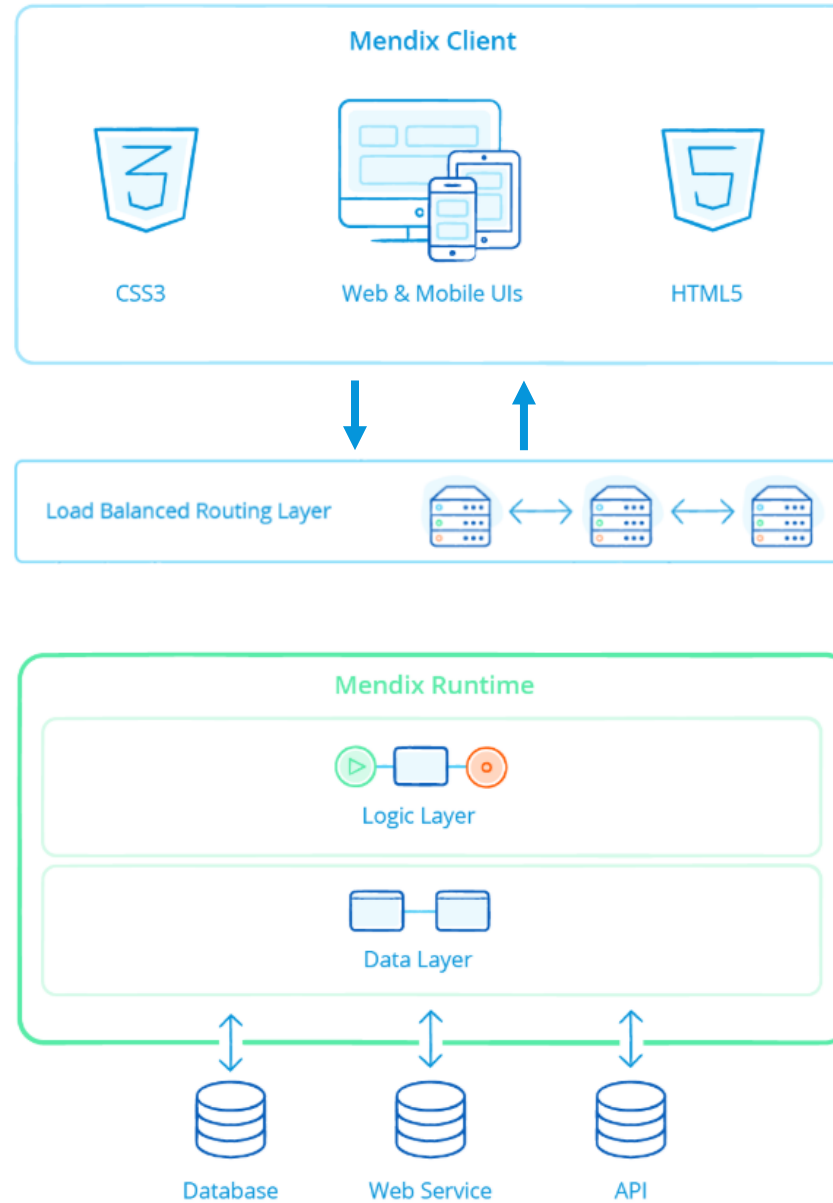


# Load Test – HTTP vs Browser

- HTTP Requests
  - More scalable
  - Can't execute client-side scripts
  - Render time can be inferred
- Browser Simulation
  - More precise
    - Client side scripts are executed
    - Includes render time
  - Less scalable

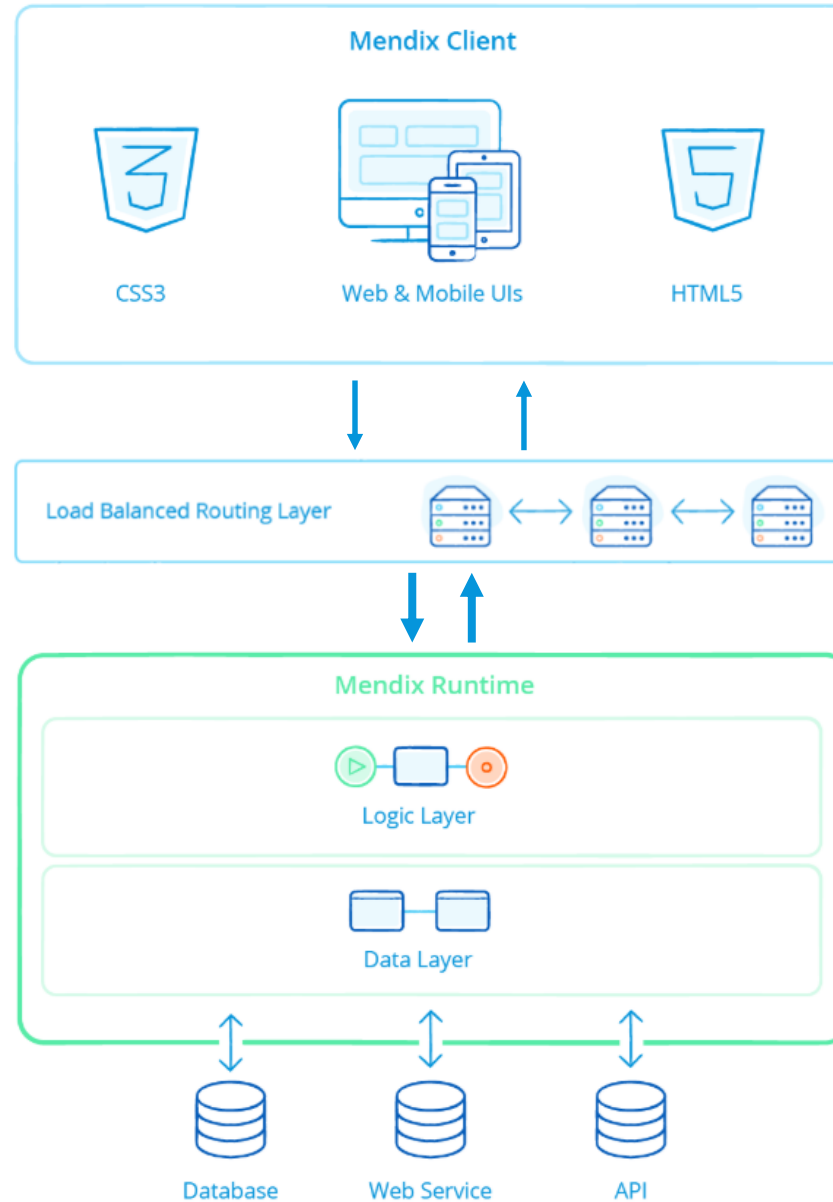
# HTTP Requests

- **Static content**
  - Fetching images or javascript files
- Dynamic Content



# HTTP Requests

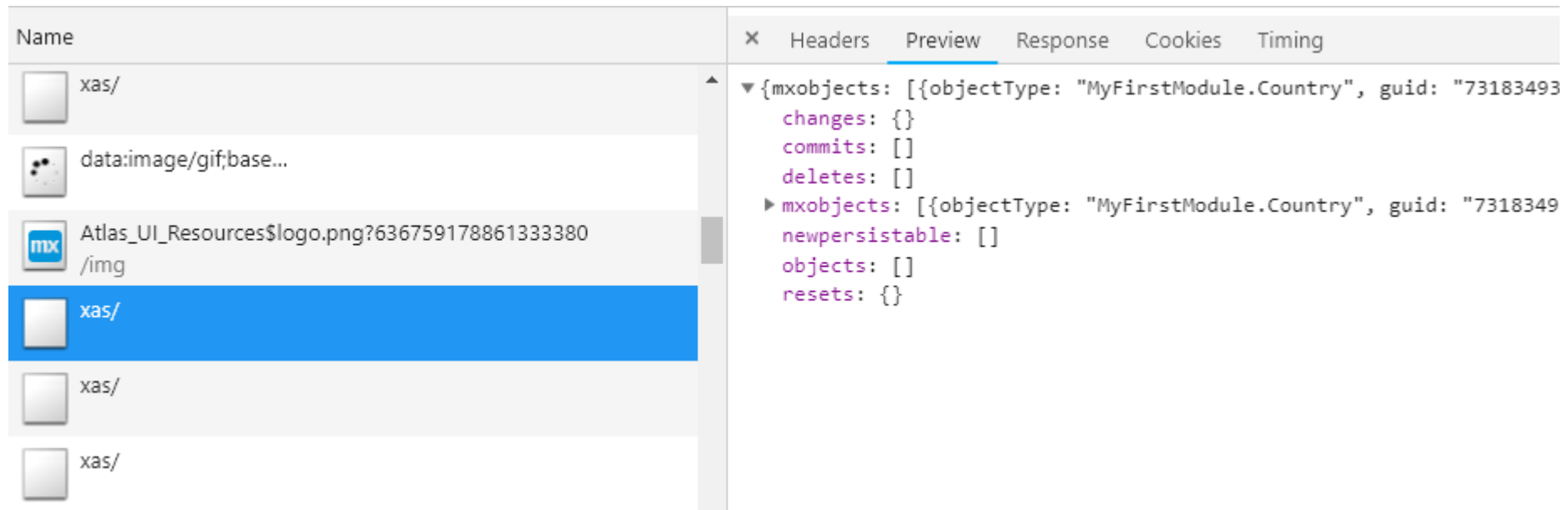
- Static content
  - Fetching images or javascript files
- **Dynamic Content**
  - **Executing microflows or fetching objects**





# Mendix XAS requests

- Dynamic requests to Mendix runtime
  - E.g. Fetch actions or microflow execution calls



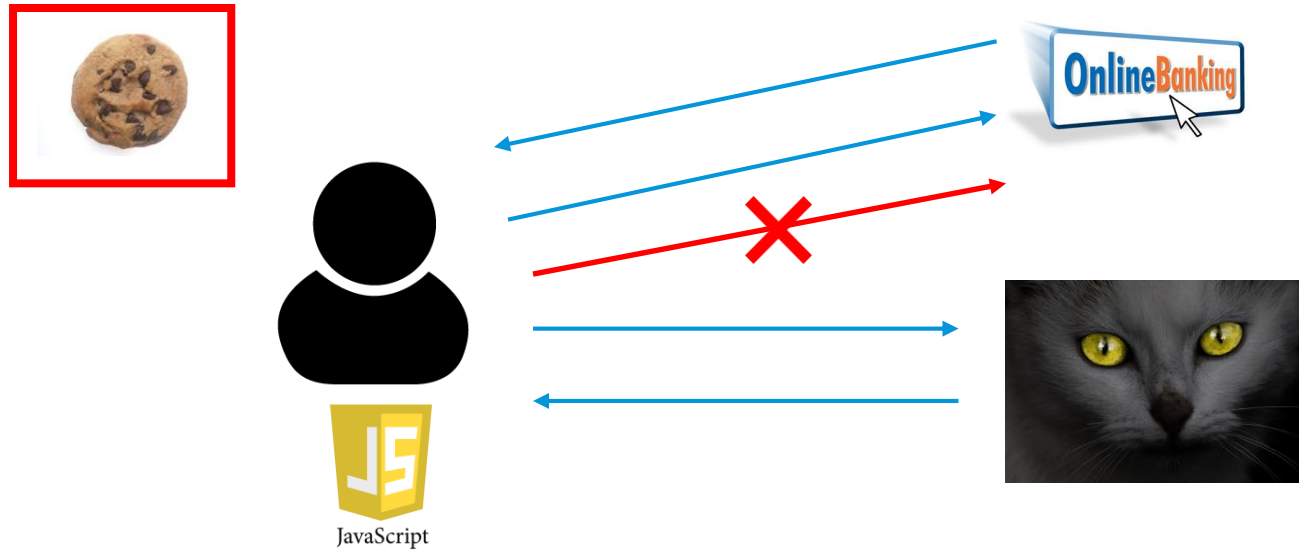
The screenshot shows a network request in a browser's developer tools. The request is to the URL 'xas/'. The response is a JSON object with the following structure:

```
{
  mxobjects: [
    {
      objectType: "MyFirstModule.Country",
      guid: "73183493",
      changes: {},
      commits: [],
      deletes: []
    },
    {
      objectType: "MyFirstModule.Country",
      guid: "7318349",
      newpersistable: [],
      objects: [],
      resets: {}
    }
  ]
}
```

# CSRF Token

- Cross-Site Request Forgery
  - Exploit web applications that trust your browser
- CSRF Tokens
  - *prevent malicious requests by third party websites to web applications that trust your browser*

# CSRF Token



<https://stackoverflow.com/questions/5207160/what-is-a-csrf-token-what-is-its-importance-and-how-does-it-work>

# Correlations – Dynamic Values

- Request – Login

```
POST https://ukservices-accp.mendixcloud.com/xas/  
  
POST data:  
{ "action": "login", "params": { "username": "optimized", "password": "Mendix1" } }
```

- Response – CSRF token

```
{ "csrftoken": "2ce7bdb6-60fa-474b-ad47-a7d38b08d998" }
```

- Request – Execute Microflow with CSRF token

```
POST https://ukservices-accp.mendixcloud.com/xas/  
  
POST data:  
{ "action": "executemicroflow", "params": { "name": "MyFirstModule.DS_GetEmployees",  
  }, "changes": {}, "objects": [], "profiledata": { "1540314231570-0": 116, "15403142316  
40314232021-3": 43 } }
```

Cookie Data:  
XASSESSIONID=997aced1-d5f2-44a9-919d-59269dc31cfd; xasid=0.0de48f14-f1e6-41bc-  
DeviceType=Desktop; Profile=Responsive

**Request Headers**  
Connection: keep-alive  
x-mx-reqtoken: 1540314232135-4  
accept: application/json  
Origin: https://ukservices-accp.mendixcloud.com  
content-type: application/json  
Referer: https://ukservices-accp.mendixcloud.com/index.html  
**x-csrf-token: 2ce7bdb6-60fa-474b-ad47-a7d38b08d998**

# JMeter



- Why
  - Free load testing tool
  - Protocol level
  - Powerful and flexible
  - Large ecosystem
  - Allows you to do large scale load testing with distributed set up
  - Lot of material online on how to use it
- Tools to create jMeter scripts (record http requests)
  - Blazemeter



# Demo app

The screenshot displays a web-based dashboard with a navigation bar at the top containing the 'mx' logo and links for Home, Dashboard, Dashboard all Items, Dashboard Optimised, and Log out. The main content area is titled 'Dashboard' and features four summary cards for regional webhop sales: US (227), EMEA (132), APAC (78), and Total (437). Below these cards is an 'Employees' section with a table listing four employees, including their roles, locations, and email addresses. A 'Feedback' button is visible on the right side of the dashboard.

Region	Webhop sales
US	227
EMEA	132
APAC	78
Total	437

Employee	Role	Location	Email
Employee1 (Japan)	Sales - Account Exec	Boston	employee1@mycompany.com
Employee2 (NL)	Sales - Pre-sales	Boston	employee2@mycompany.com
Employee3 (UK)	HR - HR	Boston	employee3@mycompany.com
Employee4 (Mexico)	Services - Delivery Manager		

# Demo 1 – Create Test Script

# Load Testing – How to check where the bottleneck is?

- Applications usually more than one system
  - Multiple applications
  - Infrastructure

## • Metrics

### • Application Server

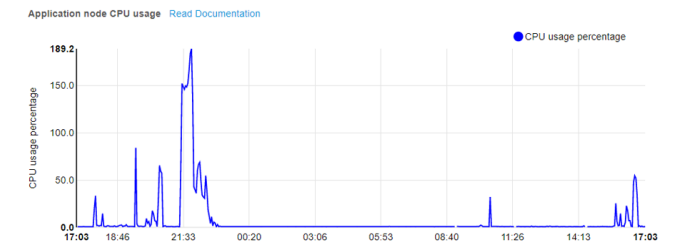
- Memory
  - e.g. Too many objects in memory
- CPU
  - e.g. Too many http requests

### • Database Server

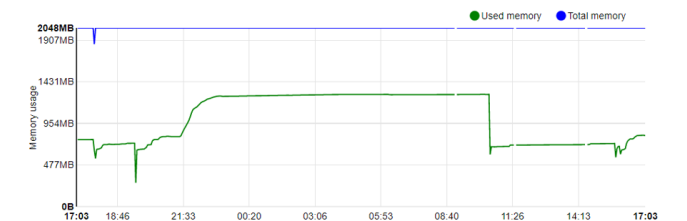
- CPU
  - Too many queries
  - Queries are too heavy and need to be optimised
  - Not enough or incorrect indexes

- <https://docs.mendix.com/developerportal/operate/trends-v4>

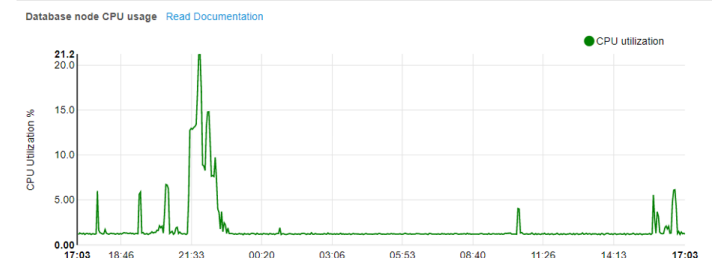
Application Node Statistics



Application node operating system memory [Read Documentation](#)



Database Node Statistics



# How many concurrent users?

- It depends – so many things could influence the final numbers
  - How many users in total?
  - What's the think time between actions?
  - How heavy are the actions the users are executing?
  - How powerful are the servers?
    - CPU, RAM, HDD/SSD, etc
  - Was the app designed correctly following performance best practices
  - Were there custom improvements made to the app/infrastructure/db?
- Start with small number of users
- Only move to next stage once everything is 100%

# What to scale/improve?

- Servers
  - Up or out
    - Out – need to be for stateless
- Threads
  - Jetty (Mendix web server) threads
  - JVM threads
  - DB Connection threads
- Database
  - Indexes
  - Denormalizing
- Application
  - Follow best practices



# Demo 2 – Non-optimized vs Optimized pages

# Results (Non-Optimized) – Why?

- With 1 Employee

- 10 xas requests from login page
  - Get session data
  - Retrieve by Id – User
  - DS\_GetEmployees
  - Retrieve by Id – Country
  - Retrieve by Id – Department
  - Retrieve by Id – Role
  - Retrieve by Id – Region
  - Retrieve by Id – Address
  - Retrieve by Id – ContactInfo (Email)
  - Retrieve by Id – County

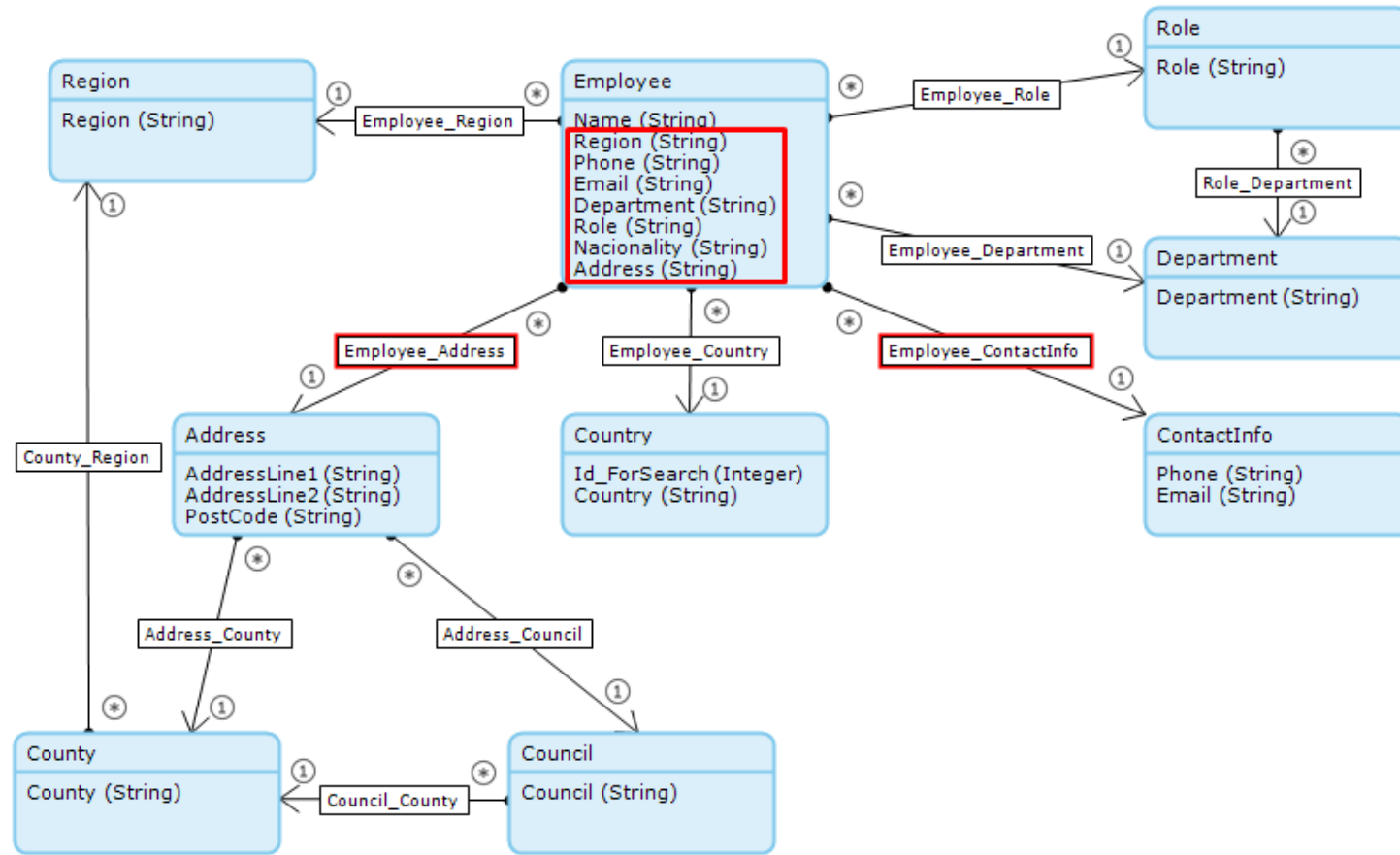
- With 2 Employees

- 17 (worst case – unique values) xas requests from login page
  - Get session data
  - Retrieve by Id – User
  - DS\_GetEmployees
  - Retrieve by Id – Country x2
  - Retrieve by Id – Department x2
  - Retrieve by Id – Role x2
  - Retrieve by Id – Region x2
  - Retrieve by Id – Address x2
  - Retrieve by Id – ContactInfo (Email) x2
  - Retrieve by Id – County x2

# Results (Optimized) – Why?

- With 1 Employee
  - 3 xas requests from login page
    - Get session data
    - Retrieve by Id – User
    - DS\_GetEmployees
- With X Employees
  - 3 xas requests from login page
    - Get session data
    - Retrieve by Id – User
    - DS\_GetEmployees

# Optimizations – Entities



# Optimizations – Pages

- Going over associations

[Employee, by microflow 'DS\_GetEmployees']

6		
{Name}		
{.../Country}		
{.../Department} - {.../Role}		
4	4	4
{.../Region}	{.../County}	{.../Email}

- Get data directly from single entity

[Employee, by microflow 'DS\_GetEmployees']

6		
{Name}		
{Nacionality}		
{Department} - {Role}		
4	4	4
{Region}	{Address}	{Email}



# More info on Performance Optimisation



## Optimize the Performance of your Apps

— Expert —

In this course you will learn which resources are available to you to quickly analyze and solve any performance issues in your apps.

## Detect and Resolve Performance Issues

Last update: Sep 24, 2018

[Edit](#)

Any application can run into performance issues. Here are a number of possible performance issues you may run into, some root causes, and finally how you can resolve them.

### Overview

The flow chart below, designed like a microflow, provides a great infrastructure for deciding how to troubleshoot and resolve performance-related issues. The rest of this article will be based on this flow.



# Create a Record

The screenshot displays a Mendix dashboard interface. At the top, a navigation bar includes the 'mx' logo and links for 'Home', 'Dashboard', 'Dashboard all items', 'Dashboard Optimised', and 'Log out'. The main dashboard area features several summary cards: 'US' with 227 Webhop sales, 'EMI' with 78 Webhop sales, and 'Total' with 437 Webhop sales. A 'Feedback' button is visible on the right side of the dashboard. In the foreground, a 'New Employee' modal form is open, containing the following fields:

- Name: Abraham Simpson
- Email: oldmanabe@simpsonfamily.com
- Address: Springfield Nursing Home
- Nacionality: Mexico (dropdown)
- Region: US (dropdown)
- Department: Marketing (dropdown)
- Role: Evangelist (dropdown)

At the bottom of the modal, there are 'Save' and 'Cancel' buttons. The background shows a table of employees with columns for name, role, location, and email.

Employee	Role	Location	Email
Employee1 (Japan)	Sales - Account Exec	Boston	
Employee10 (UK)	HR - HR	New York	employee10@mycompany.com
Employee11 (NL)	Services - Delivery Manager	New York	employee11@mycompany.com
Employee12 (Mexico)	Services - Architect		

# Correlations – POST Data

HTTP Request

Name: POST Employee data

Comments:

Basic Advanced

Web Server

Protocol [http]: https Server Name or IP: \${BASE\_URL\_1} Port Number:

HTTP Request

Method: POST Path: xas/ Content encoding:

Redirect Automatically  Follow Redirects  Use KeepAlive  Use multipart/form-data for POST  Browser-compatible headers

Parameters Body Data Files Upload

```
1 [{"action": "executemicroflow", "params": {"name": "MyFirstModule.ACT_CreateNewEmployee", "mfParams": {"Employee": {"guid": "${EMPLOYEE_GUID}"}, "validationGuids": [
"${EMPLOYEE_GUID}"]}, "changes": { "${EMPLOYEE_GUID}": {"Name": {"value": "Bart Simpson"}, "Email": {"value": "bart@eatmyshorts.com"}, "Address": {"value": "Springfield"},
"MyFirstModule.Employee_Country": {"value": "7318349394477058"}, "MyFirstModule.Employee_Region": {"value": "3096224743817217"}, "MyFirstModule.Employee_Department": {"value":
"8725724278030338"}, "MyFirstModule.Employee_Role": {"value": "7599824371187715"}}, "objects": [{"objectType": "MyFirstModule.Employee", "guid": "${EMPLOYEE_GUID}", "hash":
"${EMPLOYEE_HASH}"}, {"attributes": {"Department": {"value": "null"}, "Role": {"value": "null"}, "Name": {"value": "null"}, "Address": {"value": "null"}, "Region": {"value": "null"},
"MyFirstModule.Employee_Country": {"value": "null"}, "Email": {"value": "null"}, "MyFirstModule.Employee_ContactInfo": {"value": "null"}, "MyFirstModule.Employee_Role": {"value": "null"},
"Phone": {"value": "null"}, "MyFirstModule.Employee_Address": {"value": "null"}, "MyFirstModule.Employee_Department": {"value": "null"}, "MyFirstModule.Employee_Region": {"value": "null"},
"Nacionality": {"value": "null"}}, {"profiledata": {"1540314261697-13": 43}}]}
```

# Demo 3 – Create a Record

# Best Practices – Load Testing

- Use production set up (same environment, data, and load)
- Use realistic test scenarios
- Increase load incrementally
- Do not increase number of users until everything passes 100%
- Ramp up period and think time
- When fixing issues, change 1 thing at a time
- Monitoring on whole infrastructure
- Involve right people

# Q&A