Mx REST: The Catalyst for the API Economy

Erwin de Groot / Solution Architect / Mendix
Leonie van der Sleen / Solution Architect / Mendix
Freek Brinkhuis / Sr. Mendix Consultant / TimeSeries
Speakers

Erwin de Groot
Solution Architect
Mendix

Leonie van der Sleen
Solution Architect
Mendix

Freek Brinkhuis
Senior Mendix Consultant
TimeSeries
Agenda

1. Why use REST?
2. REST with Mendix
3. How to consume and publish REST in Mendix?
4. Customer case: Van Dorp
Why use REST?
What is REST?

1. HTTP protocol
2. Resources
3. Operations
4. Flexibility
5. Security
**JSON Example**

```json
{ "employees": [ 
    { "firstName": "John", "lastName": "Doe" },
    { "firstName": "Anna", "lastName": "Smith" },
    { "firstName": "Peter", "lastName": "Jones" }
] }
```

**XML Example**

```xml
<employees>
    <employee>
        <firstName>John</firstName> <lastName>Doe</lastName>
    </employee>
    <employee>
        <firstName>Anna</firstName> <lastName>Smith</lastName>
    </employee>
    <employee>
        <firstName>Peter</firstName> <lastName>Jones</lastName>
    </employee>
</employees>
```
Identify the Use Case for Integrations

- Extending legacy systems
- Use third party services
- Consume or Expose APIs
- Using Microservices architectures to build enterprise solutions
- Use Mendix as a mBaaS
REST with Mendix
Guiding principles

**Speed**
Be able to quickly and flexibly modify your application without impacting external dependencies.

**Collaboration**
Make it easier for others both inside and outside your organization to understand your mappings and use your API.

**Control**
You need to be able to adjust to the other side, as these are not always implemented correctly according to spec, or they may be old systems, using old practices.
Speed

Message definitions

Versioning of APIs
Collaboration

• Visual mappings to define how to convert external to internal data

• Every implemented REST API will automatically include an OpenAPI document

• Example request messages displayed in the Swagger documentation
Control

• Define import mappings to convert received messages into Mendix objects

• Define export mappings for sending object to external systems

• Mappings are reusable (use them for Web Services, REST services and for connector kit integrations)

• Works for both XML and JSON messages

• Full control in header, query and body parameters

Example import mapping
Consume REST Demo
Customer Case
Van Dorp
The Van Dorp case

One of the biggest Facility Managers of the Netherlands

Maintaining over 18,000 buildings, both office and residential buildings

Current process of handling work requires a lot of effort from both the service desk and mechanics
So what did we make?
The Mendix part

CORA is the application managing everything related to contracts, legal entities and natural persons.

TIM manages malfunction, maintenance tickets, and finances.

PANDA's main purpose is managing all data related to appointments, this also includes the automatic planning process.

NOA is responsible for creating and distributing all notifications through the platform.
The Van Dorp app landscape

Front-end

CORA

TIM

PANDA

NOA

3rd party apps

Gateway

Big data solutions
What kind of data are we talking about?

Number of objects

- Tickets: 450,000
- Buildings: 500,000
- Locations: 380,000
- Assets: 380,000
- Legal Entities:
  - Orange: 18,000
  - Green: 14,000
- Appointments: 19,000
Why REST vs SOAP?

- Speed of development
- Compatibility with modern frameworks (like REACT)
- Lower size of payloads with big datasets
- Easy modification of message definitions
- Use of swagger makes it easy for external developers and documentation
Some best practices

Authentication

Gateways – one source for multiple apps

Performance vs. completeness
Authentication

- Use JWT tokens for handling Client side authentication
- Basic authentication between Mendix apps
- API keys for Third party applications
- Profiling for access to third party resources
Gateway

- One endpoint for all applications
- Create aliases for services
- Chain requests (one request to multiple endpoints at once)
Performance vs. completeness

- Large datasets affect response times
- Flatten out your structure
- If there is no other way, try OQL
- Work with change dates, paging and limiting amount of requests
Questions