Integration based on RPC

Summary

- Classical client-server programming. The underlying socket solution remains hidden from developers.
- The channel is not only suitable for simple data transfer, remote execution of procedures also natively supported (on API level).
- Interface description
  - It is necessary to describe the remote procedures with some form of XDR (External Data Representation Standard) can help.
- Platform independence
  - Can be solved by the help of cross-platform interface descriptors

XML-RPC features

XML-RPC Basics: XML-RPC is a simple and lightweight protocol that allows applications to make remote method calls by encoding the method and complex parameters in XML format and sending them over HTTP to a remote server. The server processes the request, executes the specified method, and returns the result in XML format.

XML-RPC request and response: An XML-RPC request consists of a method call, which includes the method name and the parameters passed to it. The method name is typically represented as a string, and the parameters are specified as a list or structure. These values are then serialized into XML using predefined rules and data types.

Data Types: XML-RPC supports several basic data types, including **integers**, **booleans**, **strings**, **doubles**, **dates**, and **binary data** encoded in **Base64** encoding. Complex data structures such as arrays and structures can also be represented. The XML-RPC specification defines how these data types are encoded and decoded into XML.

XML-RPC Libraries: To integrate XML-RPC into an application, it needs an XML-RPC library specific to your programming language. Many programming languages have XML-RPC libraries available that handle the serialization and deserialization of XML-RPC messages, making it easier to work with the protocol. These libraries provide APIs to create requests, parse responses, and handle errors.

Implementation Steps: To integrate XML-RPC into your application, you generally follow these steps:

1. Sorszámozott lista elemAdd the XML-RPC library into your programming environment. This may involve installing the library or adding it as a dependency in your project.
2. Create a client that can initiate XML-RPC requests. You typically provide the method name, parameters, and endpoint URL to make a remote call. The library takes care of serializing the request into XML and sending it over HTTP.
3. Implement the XML-RPC server on the receiving end. The server listens for incoming XML-RPC requests, deserializes the XML, executes the specified method, and returns the result in XML format.
4. Handle errors and exceptions appropriately. XML-RPC provides error handling mechanisms to report exceptions or faults that occur during method execution. Both the client and server need

to handle these errors and respond accordingly.

Interoperability: One of the significant advantages of XML-RPC is its platform independency. Since XML-RPC messages are encoded in XML and transmitted over HTTP, systems written in different programming languages and running on different platforms can communicate with each other seamlessly. As long as both the client and server adhere to the XML-RPC specification, they can exchange data and invoke methods successfully.

From:
https://edu.iit.uni-miskolc.hu/ - **Institute of Information Science - University of Miskolc**

Permanent link:
https://edu.iit.uni-miskolc.hu/tanszek:oktatas:iss_t:integration_based_on_rpc_remote_process_call

Last update: **2023/05/08 19:12**