Public-API-Summary

Overview

The Scala Computing Network Simulation Public API provides programmatic access to our network simulation platform, enabling you to automate workflows, integrate with existing toolchains, and scale your simulation operations. This API transforms network simulation from a manual, interactive process into an automated, scriptable service that can become a core component of your network design and validation workflows.

What's Included

This API package contains everything you need to get started:

📚 Complete API Documentation

- **Interactive API Reference**: Full documentation of all endpoints, request/response formats, and data models
- **Swagger UI**: Interactive API explorer for testing endpoints directly in your browser
- **Quick Start Guide**: Step-by-step tutorials and examples to help you get up and running quickly
- **Code Examples**: Sample scripts demonstrating common workflows

🐍 Python Client Library



- **Pre-built Python Client**: Ready-to-use Python package for easy integration

- **Type-safe Models**: Auto-generated models matching the API schema
- **Comprehensive Examples**: Working code samples for all major operations
- **Full API Coverage**: Complete access to all API endpoints through Python classes

📋 OpenAPI Specification



- **Machine-readable Schema**: OpenAPI 3.1.2 specification for code generation
- **Generate Your Own Clients**: Use the OpenAPI spec to create clients in any language
- **API Contract**: Complete definition of all endpoints, parameters, and responses

---

Key Capabilities

The Public API provides comprehensive access to platform functionality:

Workspace Management

Organize your simulations into logical workspaces for better project management and collaboration.

Configuration Management

- Create, validate, and manage network configurations programmatically
- Define network topologies, components, and parameters
- Version control for configurations
- Validate configurations before running simulations

Model Discovery

- Browse and explore available network component models (switches, NICs, routers, applications)
- Access detailed parameter schemas for each model
- Discover model versions and capabilities

Simulation Execution

- Launch simulations programmatically
- Monitor progress in real-time
- Retrieve results automatically
- Access simulation logs and metrics
- Terminate running simulations when needed

Trace Management

- Upload and manage workload trace files (including Chakra traces)
- Automatic de-duplication of uploaded traces
- List and organize trace files
- Attach traces to simulation configurations

Template System

- Access curated, ready-to-run example configurations
- Clone templates to your workspace
- Create and share your own templates
- Demonstrate best practices for common use cases

Common Use Cases

1. Automated Regression Testing

Run automated tests via API to validate platform functionality and network configurations without manual UI interaction. Integrate test results into your

existing reporting systems.

2. Design Space Exploration

Script parameter sweeps and design of experiments (DOE) workflows to systematically explore design trade-offs. Run unattended simulations for extended periods, automatically aggregating and comparing results.

3. CI/CD Integration

Integrate network simulation into your continuous integration pipelines, automatically validating network configurations as part of your software development lifecycle.

4. Knowledge Capture and Transfer

Create reusable scripts that encapsulate best practices, enabling team members to run complex simulations correctly. Document, share, and version-control these scripts like any other code.

5. Custom Tool Development

Build specialized tools and dashboards that leverage the simulation platform, creating custom solutions tailored to your specific needs and workflows.

Getting Started

Step 1: Review the Documentation

Open the `scala-api-docs` folder and navigate to `index.html` in your web browser. This provides:
- Complete API reference
- Interactive Swagger UI for testing
- Quick start examples
- Authentication setup

Step 2: Set Up Authentication

The API uses Bearer token authentication. Configure your API key or OAuth token according to the authentication documentation.

Step 3: Try the Python Client

If you're using Python, the included client library makes integration straightforward:

```
import central_client
from central_client.rest import ApiException

# Configure your API endpoint and authentication
configuration = central_client.Configuration(
    host="https://your-platform-url/api/v1",
    access_token="your-api-token"
)

# Create an API client instance
with central_client.ApiClient(configuration) as api_client:
    # Start using the API
    workspaces_api = central_client.WorkspacesApi(api_client)
    workspaces = workspaces_api.list_workspaces()
```

Step 4: Explore Examples

Review the example scripts in the documentation to see common workflows:
- Creating a workspace
- Building a network configuration
- Running a simulation
- Retrieving results

Step 5: Build Your Workflows

Start automating your network simulation processes. Begin with simple tasks and gradually build more complex automated workflows.

API Architecture

- **RESTful Design**: Clean, intuitive REST API following OpenAPI 3.1.2 specification

- **JSON Format**: All requests and responses use JSON
- **Versioning**: Stable API versioning ensures long-term compatibility
- **Error Handling**: Clear, actionable error messages help you quickly identify and resolve issues
- **Rate Limiting**: Fair usage policies ensure system stability for all users

---

Technical Requirements

- **API Version**: 1.0.0
- **Authentication**: Bearer token (API key or OAuth)
- **Content Type**: `application/json`
- **Python Client**: Requires Python 3.9+

Support and Resources

- **API Documentation**: Complete reference available in the `scala-api-docs` folder
- **Python Client Documentation**: See the `python-client` folder for detailed Python client documentation
- **Examples**: Working code examples included in the documentation
- **Support**: Contact your Scala Computing representative for integration assistance

---

Next Steps

1. **Explore the Documentation**: Start with the introduction and quick start guide
2. **Test the API**: Use Swagger UI to try endpoints interactively
3. **Set Up Your Environment**: Configure authentication and connect to your platform instance
4. **Run Your First Simulation**: Follow the examples to create and run a simple simulation

5. **Automate Your Workflows**: Build scripts to automate your common tasks

---

Benefits of Using the API

- **Automation**: Eliminate manual, repetitive tasks
- **Integration**: Connect network simulation to your existing toolchains
- **Scalability**: Run parameter sweeps and design space exploration at scale
- **Consistency**: Codify best practices into reusable scripts
- **Efficiency**: Reduce time-to-insight with automated workflows
- **Flexibility**: Build custom solutions tailored to your specific needs

---

The Public API is designed with backward compatibility as a core principle. Once you begin building integrations, those integrations will continue to work reliably for years to come.

For questions or support, contact Scala Computing at support@scalacomputing.com