



# **P4 API Working Group Charter**

2025-01-24

# Working Group Name

P4 API Working Group

## Chairs

- Antonin Bas <[antonin@barefootnetworks.com](mailto:antonin@barefootnetworks.com)>
- Lorenzo Vicisano <[vicisano@google.com](mailto:vicisano@google.com)>

## Mailing Lists

- To subscribe, visit [http://lists.p4.org/mailman/listinfo/p4-api\\_lists.p4.org](http://lists.p4.org/mailman/listinfo/p4-api_lists.p4.org)
- To post a message to all list members, end an email to <[p4-api@lists.p4.org](mailto:p4-api@lists.p4.org)>

## Description of the Working Group

The P4 language is designed to specify the dataplane of networking devices by combining a number of core abstractions, such as parsers, tables and externs. These abstractions instantiate pipeline objects, many of which needs to be managed at runtime to configure the desired forwarding behavior in the target device. As an example, the creation and deletion of entries in match-action tables is the most notable use case of P4 object management.

Since the runtime management of P4 objects is out of the scope of the P4 language itself, this Working Group is chartered to define protocol-independent runtime APIs for P4 device management. To complement these P4 runtime APIs, this working group will also specify the relationship between P4 runtime objects and entities that appear in the switch configuration, such as ports and platform-level functions. A practical way to achieve this would be, for example, to define an [OpenConfig](#) schema specific to a switch that exposes a P4 runtime API.

A P4 runtime API should support two main pieces of functionality:

- managing the behavior of the dataplane objects described in a P4 program and retrieving information from these objects when needed. For example, in the case of match-action tables, adding entries to the table, removing them and retrieving the table content.
- (re)configuring the forwarding pipeline in a P4-programmable device, by pushing a new P4 program to the device.

We anticipate that the above functionality will require defining at least the following features:

- runtime management of P4 tables
- runtime management of Portable Standard Architecture (PSA) externs
- ability to extend the API to support vendor-specific externs
- session management, which includes updating the P4 program running on the device and packet IO between the control-plane and the device.

## Deliverables

The first item of work for the Working Group will be to define a vendor-independent, protocol-independent way for a control-plane to configure, program and control a P4 switch. To achieve this, the

Working Group will deliver at least the followings:

1. A RPC interface definition (serialization format and RPC protocol) implementing a P4 runtime API. The first RPC interface delivered by the Working Group is called P4Runtime and will target a remote controller.
2. A specification document defining the semantics of P4Runtime.
3. A specification of the configuration bindings necessary for P4 runtime control expressed as an OpenConfig schema in YANG format.
4. A working opensource implementation of 1. and 3., including a client and server libraries and bindings for the [p4lang software switch](#).