

## [SIG-NETWORK]

# Gateway API Conformance Requirements

Authors: Bowei Du, Rob Scott, <You>

### CLI Should:

1. Run tests against specified class in an existing cluster
  - a. Be able to select extended features
2. List all conformance tests
  - a. By categories
3. Detect whether or not an extended feature is enabled
4. Output some structured format (i.e. json) that we can use to generate reports for showing conformance.
5. Be consistent with ingress-controller-conformance (either update that or make gateway match)
  - a. clarify that this is intended to share code and to make it easy for implementers to conform

### Each conformance test should have a:

1. shortName (e.g. HTTPROUTE\_XYZ) so this can be referenced easily
2. Description (brief description so people can understand what the meaning is)
3. URL (to permanent link in documentation/code)
4. supportLevel : {core, extended} (note: custom will never appear in conformance, but we should have a way for implementations to extend our tests to cover custom behavior)
5. (for extended): predicate over config that says if feature is enabled or not. e.g. if (gateway.extFeature1 != nil => extFeature1 is enabled). This is a requirement for the definition of an "extended" feature.
6. test case function (func f(t \*testing.T))

### Potential Test Structure:

```
ConformanceTest{
  Name: "shortName",
  URL: GatewayDocURL("gateway#shortName"),
  Description:...
  SupportLevel: SupportExtended,
  IsEnabled: func(x) bool {},
  T: func(t *testing.T) {}
}
```

### MVP:

1. Run tests against specified class in an existing cluster
2. Only cover core features
3. Output some structured format (i.e. json) that we can use to generate reports for showing conformance.
4. Where possible, share code with ingress-controller-conformance

5. List all conformance tests
6. Test structure as defined above except for "IsEnabled" and "URL"