SIG API Machinery Subproject Meeting - Declarative APIs and Linters: Agenda & Notes

Zoom Information:

Join Zoom Meeting
https://zoom.us/j/91574890163?pwd=1uQn4MlKhatwUjk5ATti5b6l8YIIl3.1

Meeting ID: 915 7489 0163
Passcode: 77777

One tap mobile
+16699006833,,91574890163# US (San Jose)
+16694449171,,91574890163# US

Join instructions
https://zoom.us/meetings/91574890163/invitations?signature=emIgJJNCFwLZadJQNdt5Aclu5EhI
UrWMXMsKdzOa1-g

We'll meet every other Tuesday at 9AM, Pacific Time. We'll keep the meeting to 30 minutes unless there's a good reason to go longer.

This meeting agenda gets sent to the SIG each Monday afternoon for people to propose agenda topics, and each Tuesday afternoon for visibility of the upcoming agenda.

When adding items:
- If your item is about a design/pr/issue, give some indication of what you're hoping to achieve: is it informational? To identify blockers? To get advice / help / mentorship? To get a decision?

Key Objectives

1. **Simplify API Development:** Replace complex handwritten Go code with declarative expressions
2. **Improve Review Process:** Reduce cognitive load on API reviewers through automated validation
3. **Unify Experience:** Bridge the gap between Kubernetes native types and CRDs

4. **Enable Automation:** Facilitate automated API quality checks and documentation generation

Slack Chatroom

- For discussion topics related to this meeting, we use the
  - [# sig-api-machinery-dev-tools slack channel](#)

Youtube Recordings

- Playlist of all meeting recordings - https://www.youtube.com/playlist?list=PL--jPUI8HA10IosS6xSbUstXIG05IiHkz

Agenda Backlog (please add your name, email or slack when adding an item)

-

Oct 7, 2025 (Next Meeting)

- Meeting host
  - Aaron Prindle
- Go through linting wishlist
  - Go into k/k
  - Depend on DV for k/k
  - Upcoming
  - From @sttts on Slack
    - Be explicit about optional and required.
    - For enums, document the possible values.
    - Document defaults.
    - MinLength on non-optional strings.
    - Make the descriptions start with the JSON field name, not with the Go field name.
    - Pointers for optional fields.
- How to contribute to DV
- if[Enabled|Disabled] format - DV vs CRD usage
- Parsing DV tags (chaining syntax, etc.)
- Immutability, ratcheting, and feature gating

Sep 23, 2025

- Meeting Host
  - Aaron Prindle

- Agenda
  - Declarative Validation Overview and Roadmap
    - Slides
      - 🟧 [PUBLIC] New Kubernetes Feature - Declarative Validation /…
      - v1.33
        - validation-gen framework core completed
        - Early set of base tags developed in a dev-branch
        - One validation migration - ReplicationController.Replicas
          - +k8s:minimum tag
      - v1.34
        - Validation-gen framework complex cases workstream
        - Tags extended to include +k8s:unionMember and +k8s:zeroOrOneOfMember
        - Ratcheting implementation
        - One validation migration -CertificateSigningRequestStatus.Conditions
          - Validation: at most one of terminal states Approved OR Denied
    - validation-gen dev-branch - http://github.com/jpbetz/validation-gen
      - ^ for getting involved and getting into discussion there
      - **Aaron Prindle to tag help-wanted**
      - **Joe Betz - do folks want to hear about immutability, ratcheting or feature gating here**
        - **joel.speed@hotmail.co.uk +1 🎊**
        - **sbueringer@gmail.com +1**
    - Joel Speed - stability levels?
      - We have a graduation process for tags where tags graduate across stability levels
      - KEP update incoming around this
      - @jpbetz - discussion around go/no-go. We have informal agreement to go forward here and support DV Only validations
    - Current state of project
      - KEP: https://github.com/kubernetes/enhancements/tree/master/keps/sig-api-machinery/5073-declarative-validation-with-validation-gen
    - Docs
      - https://kubernetes.io/docs/reference/using-api/declarative-validation/
      - https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api_changes.md
      - https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md
    - Codifying validation best practices

- Recent docs PR:
  https://github.com/kubernetes/community/pull/8616
    - Migration Plan and Call To Action
      - Asking folks that are working on new APIs with net new validation code to use Declarative Validation where possible
      - Also asking folks to begin migrating simple tags
- Kube api linter
  - Recent updates
    - Linter aimed at API types and easing API review
    - 22 rule implemented, 15 open issues for other rules (as of 23rd Sep)
    - Enabled both ssatags (listType) and conditions linters on K/K
    - Need to prio:
      - Updating docs for golangci-lint v2
      - Creating an entrypoint that we can build standalone
        - Module & plugin atm
      - Publishing a v0.1.0
      - Investigating additional +k8s marker implications
    - Usage: Cluster api, openshift, k/k
    - @jpbetz - +1 for optional and default linting 🙏. Difficult case for reviewers
- Ongoing API Conventions and Validation Best Practices Discussions
  - Immutability
  - Ratcheting
- Recent conversations:
  - IDL Tag Migration Strategy: future of existing markers like +optional and whether they'll be deprecated in favor of new declarative validation markers
  - Controller-gen Integration & Standardization: Discuss feature parity between the new k8s IDL tags, what's currently available to CRD developers through kubebuilder, and standardizing IDL tag semantics between k/k and controller-gen
- joel.speed@hotmail.co.uk - core API types discussion
  - Missing opinion of CRD authors
  - Stefan, Bryce, Joel have experience here
  - Ex: ifEnabled - w/ 4 types need payload
    - In CRD doesn't make sense, gateway API. identical validation in open-shift
    - Q: w/ all these discussion, how do we make sure the markers work for types BUT we consider CRDs
    - "This is how it works" -> 🥲 w/ CRDs
      - Maybe identical is not feasible but close as possible
    - Joe Betz - some tags are easy, 1:1 w/ Open API
      - Enum is similar

- - ○ Union -> oneOf is similar
    - ○ ifEnabled and feature gates
  - ● Joe Betz - in apiserver doessn't go if feature -> then
    - ○ We do two passes
      - ■ In use detection
      - ■ If feature gate is on OR in use -> can use feature
      - ■ Gate in the sense obj is in once in
        - ● ^ complex mechanism
- ○ Yongrui Lin - kube-apilinter SSA tags in k/k
  - ■ DV has own SSA tags w/ k8s: prefix
    - ● Both exist long-term
    - ● Opinion, support both in kube-apilinter
  - ■ Eventually we want to deprecate old tags
  - ■ Kube-apilinter - don't use kubebuilder, use upstream
    - ● End goal is upstream versions (+* -> +k8s:*)
    - ● Don't currently have marker duplication
  - ■ Better to support both now, we are about to open gate for DV Only
    - ● New API use +k8s:* prefix tags, if we have linter that covers this - need to extend linter
  - ■ Bryce - can forbid certain markers
    - ● Maybe exclude linting existing APIs and enforce rule where not +k8s: this is forbidden
  - ■ joel.speed@hotmail.co.uk - tag chaining is new, kube-apilinter parsing doesn't handle this new chaining currently
    - ● How will this work?
      - ○ +k8s:ifEnabled()=+k8s:listType….
        - ■ ^ SSA tag linting
  - ■ Yongrui Lin - different parsing that controller-tools w/ chaining
    - ● Gengo parse chaining
    - ● Controller-tools is different
    - ● Kube-apilinter similar to controller-tools
    - ● Joe Betz - for DV we have a tag w/ a consistent style
      - ○ Not mixing w/ other tags and structure
      - ○ Maybe extend to kubebuilder?
      - ○ In gengo we have a parser for this
      - ○ Lalit Chauhan - use gengo to parse chained tag (new concept). Syntax understood by gengo mainly atm. Payload application. Existing one is key value pair
      - ○ Bryce - parsing logic in gengo, we should be able to implement tag parsing in this case
        - ■ Look at suffix, we can do DV tag parsing
          - ● If not: controller-tools, kubebuilder tags

- ○ Joe Betz - let us know how that impl goes, we have namespacing to make a first pass attempt this
  - ○ joel.speed@hotmail.co.uk - chaining
    - ■ **eachKey (delta)**
    - ■ eachVal
    - ■ **ifEnabled|Disabled (some diffs)**
    - ■ +k8s:subfield
    - ■ +k8s:item
      - ● CEL
  - ○
- ● Bryce - wishlist items around CRD type linting rules
  - ○ Go through wishlist and see what could be done for both
  - ○ Get linters in place, align DV work
- ● Where to chat async after this meeting
  - ○ Slack
  - ○ SIG API Machinery dev tools
- ● Abdul - docs, how to get started here
  - ○ Chat async around work here in sig api machinery dev tools
- ● ASYNC CHAT LOCATION
  - ○ [# sig-api-machinery-dev-tools Slack Channel](#)
- ● AIs
  - ○ Go through linting wishlist
    - ■ Go into k/k
    - ■ Depend on DV for k/k
    - ■ upcoming
  - ○ How to contribute to DV
  - ○ ifEnabled format
  - ○ Parsing DV tags