

OpenAct Design Spec (v1.0)

0. 目标与范围

- **目标**：用最小的 OpenAPI 子集 + 少量 **x-*** 扩展，描述"单个 API 动作 (Action)"，并通过统一 Runner 执行；鉴权与过期刷新通过集中配置 (provider 级) 统一管理。
 - **范围**：本规范定义配置结构、合并顺序、执行时序、失败语义、可观测性与测试基线；**不包含**具体编程语言实现。
-

1. 核心概念与对象模型

1.1 Action (动作)

- 一个 **Action** MUST 由 **单个 OpenAPI operation** 表示：即一个 **path + method**。
- 一个 Action 文件 (YAML/JSON) MUST 只包含 **一个 paths/<path>/<method>**。
- Action 使用的扩展字段 MUST 以 **x-** 开头 (见 §3)。

1.2 Provider (提供商)

- Provider 指 API 域或服务 (如 **slack.com**、**api.github.com**、**googleapis.com**)。
- Provider 级默认配置按"**主域名**"匹配 (见 §2)，用于统一鉴权与执行规则。

1.3 TRN (Tool Resource Name)

- **connection_trn** 是跨模块引用认证连接的标识字符串。格式不强制，**但** SHOULD 全局唯一且可被 AuthFlow 解析。
-

2. 配置层次与合并顺序

2.1 四层配置

1. **Provider Auth Defaults (鉴权默认)** — **provider-auth-defaults.json**
 - 描述该 Provider 的标准鉴权注入、过期判断、刷新策略与失败语义。
2. **Provider Defaults (执行默认, 可选)** — **provider-defaults.json**
 - 描述该 Provider 的通用执行规则 (分页、重试、错误路径、成功判定、默认超时等)。
3. **Action 文档 (OpenAPI 单 operation)**
 - 仅描述该动作本身；应仅指定 **x-auth.connection_trn**，其他可继承上游默认。
4. **Sidecar 覆盖 (可选)**
 - 针对某个 Action 的定点覆盖 (如租户/环境特例)，优先级最高。

2.2 合并优先级 (从低到高)

1. Provider Auth Defaults / Provider Defaults
2. Action 文档的同名 **x-***
3. Sidecar 覆盖

3. 扩展字段（x-* 白名单）

下述 x-* 字段 MUST 出现在 **operation 对象**（`paths/<path>/<method>`）层级。

3.1 x-auth（鉴权与过期刷新）

```
x-auth:
  connection_trn: string # MUST. 指向 AuthFlow 中
  的连接
  scheme: bearer|oauth2|apikey|basic|service_account # SHOULD. 若缺省由
  Provider Auth Defaults 供给
  injection: # MUST
    type: jsonada # 固定：使用 jsonada 引擎
    mapping: string # MUST. stepflow-dsl 语
    法表达式，输出为将注入的 headers 或 query 映射
  expiry: # MAY
    source: field|header|none # default: field
    field: string # 如 "$expires_at" (当
    source=field)
    header: string # 如 "X-Token-
    Expires" (当 source=header)
    clock_skew_ms: integer ≥ 0 # default: 30000
    min_ttl_ms: integer ≥ 0 # default: 0
  refresh: # MAY
    when: proactive|on_401|proactive_or_401 # default:
    proactive_or_401
    max_retries: integer ≥ 0 # default: 1
    cooldown_ms: integer ≥ 0 # default: 0
  failure: # MAY
    reauth_error_code: string # default: "E_AUTH"
    bubble_provider_message: boolean # default: true
```

jsonada 求值上下文（固定）：

- `$access_token`, `$expires_at`（来自 AuthFlow 连接）
- `$ctx`（运行上下文，可选）
- 表达式 MUST 产出一个对象（例如注入 headers 映射）。

3.2 x-retry（重试）

```
x-retry:
  on_status: [integer, ...] # default:
  [429,500,502,503,504]
  respect_retry_after: boolean # default: true
  strategy: exponential|linear|none # default: exponential
  base_ms: integer ≥ 1 # default: 400
```

<code>max_retries: integer ≥ 0</code>	<code># default: 5</code>
<code>jitter: none full</code>	<code># default: full</code>

3.3 x-pagination (分页)

<code>x-pagination:</code>	
<code>strategy: none cursor pageToken link</code>	<code># default: none</code>
<code>cursor_param: string</code>	<code># 如 "cursor" 或</code>
<code>"pageToken"</code>	
<code>cursor_path: string</code>	<code># jsonada, 定位下一页游标</code>
<code>items_path: string</code>	<code># jsonada, 定位本页数据数</code>
<code>stop_when: string</code>	<code># jsonada 布尔表达式 (可</code>
<code>选, true=停止)</code>	

3.4 结果与错误

<code>x-ok-path: string null</code>	<code># 成功判定路径; null 代表</code>
<code>用 HTTP 2xx</code>	
<code>x-error-path: string</code>	<code># 提取错误信息的路径;</code>
<code>Provider Defaults 可给出</code>	
<code>x-output-pick: string</code>	<code># jsonada, 裁剪输出体 (可</code>
<code>选)</code>	
<code>x-timeout-ms: integer ≥ 1</code>	<code># 请求超时 (ms)。</code>
<code>default: 15000 (建议)</code>	

4. Provider 层配置

4.1 provider-auth-defaults.yaml (必备)

- 顶层键 `MUST` 是域名 (主机名), 如 `api.github.com`、`slack.com`。
- 值 `MUST` 是对象, 其中含 `x-auth` 模板 (字段见 §3.1, 不含 `connection_trn`)。

4.2 provider-defaults.yaml (可选)

- 顶层键 `MUST` 是域名。
- 值可包含: `x-retry`、`x-pagination`、`x-ok-path`、`x-error-path`、`x-timeout-ms` 等 (见 §3.2–§3.4)。

5. 单 Action 文档 (OpenAPI 最小子集)

5.1 必需字段

- `openapi: 3.0.x` 或 `3.1.x`

- `servers[0].url`: 基础 URL
- `paths/<path>/<method>`: 唯一 operation
- `operationId`: 稳定唯一 (建议 `<provider>.<resource>.<action>`)
- `responses`: 至少一个 2xx
- `security`: 建议 `[]` (鉴权通过 `x-auth` 注入)

5.2 最小示例 (只指定 TRN)

```
openapi: 3.0.3
info: { title: GitHub Get User, version: 1.0.0 }
servers: [ { url: https://api.github.com } ]

paths:
  /user:
    get:
      operationId: github.user.get
      security: []
      responses:
        '200': { description: OK }

      x-auth:
        connection_trn: "trn:authflow:tenant123:connection/github-user123"
        # 其他 x-* 全部继承 provider-defaults 与 provider-auth-defaults
```

6. Sidecar 覆盖 (可选)

- Sidecar 覆盖 MUST 以 **operationId** 为定位键。
- Sidecar 的结构 MUST 与 `x-*` 字段一致 (仅出现需要覆盖的键)。
- 覆盖优先级 MUST 高于 Action 文档。

7. Runner 执行时序 (规范)

1. 解析 Action → 解析 provider → 合并四层配置。
2. 调用 AuthFlow 获取凭据；若需刷新，按 `expiry` + `refresh` 策略执行。
3. 使用 `jsonada` 执行 `x-auth.injection.mapping`，生成 headers/query。
4. 发请求，应用 `x-timeout-ms` 与 `x-retry`。
5. 遇 401/invalid_token → 若策略允许 → 刷新并重放一次。
6. 判定成功/错误: `x-ok-path` / `x-error-path`。
7. 若有 `x-pagination` → 循环获取，按 `items_path` 聚合。
8. 若有 `x-output-pick` → 最终裁剪输出。

8. 测试与合规

- **Lint** : Action 文件 MUST 通过 OpenAPI 校验；仅 1 个 operation；所有 `x-*` 符合本规范。

- **契约测试**：每个 Action 至少两个用例（成功 / 错误），分页 Action 须有分页用例。
- **Golden 回放**：建议保存脱敏响应，回放优先。

9. 总结

- 每个 Action = 一个最小 OpenAPI 文档，仅需指定 `connection_trn`。
- Provider 级默认集中维护 `x-auth` 与通用规则。
- Runner 按四层配置合并执行，使用 **统一的 jsonada** 求值语义。
- Sidecar 可做特例覆盖，不破坏 Action 文件。

附录 A：配置示例

A.1 Provider Auth Defaults 示例

```
# provider-auth-defaults.yaml
api.github.com:
  scheme: "oauth2"
  injection:
    type: "jsonada"
    mapping: |
      {
        "Authorization": "{% 'Bearer ' & $access_token %}",
        "Accept": "application/vnd.github+json"
      }
  expiry:
    source: "field"
    field: "$expires_at"
    clock_skew_ms: 30000
  refresh:
    when: "proactive_or_401"
    max_retries: 1

slack.com:
  scheme: "oauth2"
  injection:
    type: "jsonada"
    mapping: |
      {
        "Authorization": "{% 'Bearer ' & $access_token %}"
      }
```

A.2 Provider Defaults 示例

```
# provider-defaults.yaml
api.github.com:
  x-retry:
```

```

on_status: [429, 500, 502, 503, 504]
strategy: "exponential"
base_ms: 400
max_retries: 3
x-timeout-ms: 15000
x-ok-path: null
x-error-path: "$.message"

```

A.3 复杂 Action 示例

```

openapi: 3.0.3
info: { title: GitHub List Repos, version: 1.0.0 }
servers: [ { url: https://api.github.com } ]

paths:
  /user/repos:
    get:
      operationId: github.repos.list
      security: []
      parameters:
        - name: per_page
          in: query
          schema: { type: integer, default: 30 }
        - name: page
          in: query
          schema: { type: integer, default: 1 }
      responses:
        '200': { description: OK }

      x-auth:
        connection_trn: "trn:authflow:tenant123:connection/github-
user123"

      x-pagination:
        strategy: pageToken
        cursor_param: "page"
        cursor_path: "{% $.next_page %}"
        items_path: "{% $ %}"
        stop_when: "{% $.length < per_page %}"

      x-output-pick: "{% $.map(function($repo) { { id: $repo.id, name:
$repo.name, full_name: $repo.full_name } }) %}"

```

A.4 Sidecar 覆盖示例

```

# sidecar-overrides.yaml
github.repos.list:
  x-retry:

```

```
max_retries: 5
x-timeout-ms: 30000
```

附录 B : jsonada 表达式示例

B.1 基础认证注入

```
{
  "Authorization": "{% 'Bearer ' & $access_token %}"
}
```

B.2 复杂头部注入

```
{
  "Authorization": "{% 'Bearer ' & $access_token %}",
  "Accept": "application/vnd.github+json",
  "User-Agent": "manifest/1.0",
  "X-Request-ID": "{% $ctx.execution_id %}"
}
```

B.3 条件性注入

```
{% $ctx.method = "POST" ?
{
  "Authorization": "{% 'Bearer ' & $access_token %}",
  "Content-Type": "application/json"
} :
{
  "Authorization": "{% 'Bearer ' & $access_token %}"
}
%}
```

B.4 分页游标提取

```
{% $.pagination.next_cursor %}
```

B.5 数据数组提取

```
{% $.data.items %}
```

B.6 输出裁剪

```
{% $.map(function($item) {  
  {  
    id: $item.id,  
    name: $item.name,  
    created_at: $item.created_at  
  }  
}) %}
```

附录 C：错误码规范

C.1 标准错误码

- **E_AUTH**: 认证失败
- **E_TIMEOUT**: 请求超时
- **E_RETRY_EXHAUSTED**: 重试次数耗尽
- **E_PAGINATION**: 分页错误
- **E_JSONADA**: jsonada 表达式执行错误
- **E_PROVIDER**: Provider 配置错误

C.2 错误响应格式

```
error:  
  code: "E_AUTH"  
  message: "Authentication failed"  
  details:  
    provider: "api.github.com"  
    operation_id: "github.user.get"  
    connection_trn: "trn:authflow:tenant123:connection/github-user123"
```

文档版本 : v1.0

最后更新 : 2025年09月

作者 : OpenAct Team