Table of Contents

- 소개
- 공통 사항
  - Domain
  - Common Response
- 1. 개요
- 2. Series API
  - 2-1. 시리즈 작성
    - 시리즈 작성
    - 시리즈 작성 성공
  - 2-2. 시리즈 조회
    - 시리즈 조회
    - 시리즈 조회 성공
    - 시리즈 리스트 조회
    - 시리즈 리스트 조회 성공
  - 2-3. 시리즈 수정
    - 시리즈 수정
    - 시리즈 수정 성공
  - 2-4. 시리즈 삭제
    - 시리즈 삭제
    - 시리즈 삭제 성공
- 3. Post API
  - 3-1. 게시글 작성
    - 게시글 작성
    - 게시글 작성 성공
  - 3-2. 게시글 조회
    - 게시글 조회
    - 게시글 조회 성공
  - 3-3. 게시글 수정
    - 게시글 수정
    - 게시글 수정 성공
  - 3-4. 게시글 삭제
    - 게시글 삭제
    - 게시글 삭제 성공
- 4. Comment API
  - 4-1. 댓글 등록
    - 댓글 등록
    - 댓글 등록 성공
  - 4-2. 댓글 조회
    - 댓글 조회
    - 댓글 조회 성공
    - 댓글 리스트 조회
    - 댓글 리스트 조회 성공
  - 4-3. 댓글 삭제
    - 댓글 삭제
    - 댓글 삭제 성공

API Document

Version 1.0

소개

Mylog API

공통 사항

API에 관계없이 아래 사항을 지켜주셔야 합니다.

Domain

환경	Domain
개발서버	`https://mylog.cf:8081`

Common Response

---

1. 개요

---

2. Series API

2-1. 시리즈 작성

시리즈 작성

Request Header

Path	Type	Description
`name`	`String`	시리즈 이름

Path	Type	Description
`isSuccess`	`Boolean`	요청 성공 여부
`status`	`Number`	응답 상태
`code`	`Number`	응답 코드
`message`	`String`	응답 메시지
`timestamp`	`String`	api 호출 일시

시리즈 작성 성공

Request Example

```
POST /series HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Content-Length: 38
Host: mylog.cf

{
  "name" : "새로운 시리즈"
}
```

Response Example

```
HTTP/1.1 201 Created
Content-Type: application/json;charset=UTF-8
Content-Length: 173

{
  "isSuccess" : true,
  "status" : 201,
  "code" : 203,
  "message" : "시리즈 등록에 성공하였습니다.",
  "timestamp" : "2021-11-04T21:32:46.953+00:00"
}
```

2-2. 시리즈 조회

시리즈 조회

Request Header

Path	Type	Description
`isSuccess`	`Boolean`	요청 성공 여부
`status`	`Number`	응답 상태
`code`	`Number`	응답 코드
`message`	`String`	응답 메시지
`result.commentId`	`Number`	댓글 번호
`result.postId`	`Number`	게시글 번호
`result.commentWriter`	`String`	댓글 작성자
`result.commentContent`	`String`	댓글 내용
`result.commentCreatedAt`	`String`	댓글 작성일
`result.commentUpdatedAt`	`String`	댓글 수정일
`timestamp`	`String`	api 호출 일시

시리즈 조회 성공

Request Example

```
GET /series/6 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Host: mylog.cf
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 239

{
  "isSuccess" : true,
  "status" : 200,
  "code" : 204,
  "message" : "시리즈 조회에 성공하였습니다.",
  "result" : {
    "id" : 6,
    "name" : "알고리즘"
  },
  "timestamp" : "2021-11-04T21:32:47.072+00:00"
}
```

시리즈 리스트 조회

Request Header

Path	Type	Description
`isSuccess`	`Boolean`	요청 성공 여부
`status`	`Number`	응답 상태
`code`	`Number`	응답 코드
`message`	`String`	응답 메시지
`result`	`Array`	시리즈 조회 결과 리스트
`result.[].id`	`Number`	시리즈 번호
`result.[].name`	`String`	시리즈 이름
`timestamp`	`String`	api 호출 일시

시리즈 리스트 조회 성공

Request Example

```
GET /series/ HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Host: mylog.cf
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 296

{
  "isSuccess" : true,
  "status" : 200,
  "code" : 204,
  "message" : "시리즈 조회에 성공하였습니다.",
  "result" : [ {
    "id" : 6,
    "name" : "알고리즘"
  }, {
    "id" : 15,
    "name" : "개발기록"
  } ],
  "timestamp" : "2021-11-04T21:32:47.167+00:00"
}
```

2-3. 시리즈 수정

시리즈 수정

Request Header

Table 1. /series/{id}

Parameter	Description
`id`	수정 요청할 시리즈 ID

Path	Type	Description
`name`	`String`	시리즈 이름

Path	Type	Description
`isSuccess`	`Boolean`	요청 성공 여부
`status`	`Number`	응답 상태
`code`	`Number`	응답 코드
`message`	`String`	응답 메시지
`timestamp`	`String`	api 호출 일시

시리즈 수정 성공

Request Example

```
PATCH /series/6 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Content-Length: 38
Host: mylog.cf

{
  "name" : "수정된 시리즈"
}
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 173

{
  "isSuccess" : true,
  "status" : 204,
  "code" : 201,
  "message" : "시리즈 수정에 성공하였습니다.",
  "timestamp" : "2021-11-04T21:32:46.739+00:00"
}
```

2-4. 시리즈 삭제

시리즈 삭제

Request Header

Table 2. /series/{id}

Parameter	**Description**
`id`	삭제 요청할 시리즈 ID

Path	**Type**	**Description**
`isSuccess`	`Boolean`	요청 성공 여부
`status`	`Number`	응답 상태
`code`	`Number`	응답 코드
`message`	`String`	응답 메시지
`timestamp`	`String`	api 호출 일시

시리즈 삭제 성공

Request Example

```
DELETE /series/6 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Host: mylog.cf
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 173

{
  "isSuccess" : true,
  "status" : 204,
  "code" : 202,
  "message" : "시리즈 삭제에 성공하였습니다.",
  "timestamp" : "2021-11-04T21:32:46.551+00:00"
}
```

3. Post API

3-1. 게시글 작성

게시글 작성

Request Header

Path	Type	Description
`seriesId`	`Number`	시리즈 번호
`title`	`String`	게시글 제목
`content`	`String`	게시글 내용

Path	Type	Description
`isSuccess`	`Boolean`	요청 성공 여부
`status`	`Number`	응답 상태
`code`	`Number`	응답 코드
`message`	`String`	응답 메시지
`timestamp`	`String`	api 호출 일시

게시글 작성 성공

Request Example

```
POST /posts HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Content-Length: 90
Host: mylog.cf

{
  "seriesId" : 6,
  "title" : "새로운 게시글",
  "content" : "반가워요!"
}
```

Response Example

```
HTTP/1.1 201 Created
Content-Type: application/json;charset=UTF-8
Content-Length: 173

{
  "isSuccess" : true,
  "status" : 201,
  "code" : 202,
  "message" : "게시글 등록에 성공하였습니다.",
  "timestamp" : "2021-11-04T21:32:46.038+00:00"
}
```

3-2. 게시글 조회

게시글 조회

Parameter	Description
`seriesId`	시리즈 번호
`hasSeries`	시리즈 유무
`search`	제목/내용 검색어
`page`	페이지 번호
`size`	페이지 사이즈

Path	Type	Description
`isSuccess`	`Boolean`	요청 성공 여부
`status`	`Number`	응답 상태
`code`	`Number`	응답 코드
`message`	`String`	응답 메시지
`page.currentPage`	`Number`	현재 페이지 번호
`page.pageSize`	`Number`	페이지 사이즈
`page.totalPages`	`Number`	전체 페이지 수
`page.totalElements`	`Number`	전체 요소 수
`result`	`Array`	게시글 조회 결과 리스트
`result.[].seriesId`	`Number`	시리즈 번호
`result.[].seriesName`	`String`	시리즈 이름
`result.[].postId`	`Number`	게시글 번호
`result.[].postTitle`	`String`	게시글 제목
`result.[].postContent`	`String`	게시글 내용
`result.[].postCreatedAt`	`String`	게시글 작성일
`result.[].postUpdatedAt`	`String`	게시글 수정일
`result.[].commentCount`	`Number`	댓글 수
`timestamp`	`String`	api 호출 일시

게시글 조회 성공

Request Example

```
GET /posts/?seriesId=6&hasSeries=&search=&size=10&page=1&hasSeries=&search= HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Host: mylog.cf
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 7747

{
  "isSuccess" : true,
  "status" : 200,
  "code" : 202,
  "message" : "게시글 조회에 성공하였습니다.",
  "page" : {
    "currentPage" : 0,
    "pageSize" : 10,
    "totalPages" : 1,
    "totalElements" : 4
  },
  "result" : [ {
    "seriesId" : 6,
    "seriesName" : "알고리즘",
    "postId" : 11,
    "postTitle" : "JAVA 코딩테스트 문법 [2편]",
    "postContent" : "\n[Collection]\nlist -> Array\n리스트를 배열로 바꾸는 법\n타입이 같다면 toArray를 사용하면 된다
    "postCreatedAt" : "2021-11-05T03:56:27",
    "postUpdatedAt" : "2021-11-05T03:57:28",
    "commentCount" : 0
  }, {
    "seriesId" : 6,
    "seriesName" : "알고리즘",
```

```
    "postId" : 10,
    "postTitle" : "JAVA 코딩테스트 문법 [1편]",
    "postContent" : "[문자열]\n\n[문자열 비교]\nstr1.compareTo(str2) //int 반환\n\nstr1이 str2보다 사전순으로 뒤인
    "postCreatedAt" : "2021-11-05T03:54:15",
    "postUpdatedAt" : "2021-11-05T03:54:52",
    "commentCount" : 0
  }, {
    "seriesId" : 6,
    "seriesName" : "알고리즘",
    "postId" : 9,
    "postTitle" : "순열과 조합",
    "postContent" : "순열\n- 중복 X 순서 O\n- 중복순열에서 같은 값(중복)만 빼준것 (ex 1,1 3,3 5,5)\n\n중복순열\n- 중
    "postCreatedAt" : "2021-11-05T03:49:03",
    "postUpdatedAt" : "2021-11-05T03:49:53",
    "commentCount" : 1
  }, {
    "seriesId" : 6,
    "seriesName" : "알고리즘",
    "postId" : 2,
    "postTitle" : "이진 탐색",
    "postContent" : "순차탐색 \n1리스트 안에 있는 특정한 데이터를 찾기 위해 앞에서부터 데이터를 하나씩 차례대로 확인하는
    "postCreatedAt" : "2021-11-02T13:37:24",
    "postUpdatedAt" : "2021-11-05T02:52:35",
    "commentCount" : 2
  } ],
  "timestamp" : "2021-11-04T21:32:46.339+00:00"
}
```

3-3. 게시글 수정

게시글 수정

Request Header

Table 3. /posts/{id}

Parameter	Description
`id`	수정 요청할 게시글 ID

Path	Type	Description
`seriesId`	`Number`	시리즈 번호 (없는 번호일 경우 미분류)
`title`	`String`	게시글 제목
`content`	`String`	게시글 내용

Path	Type	Description
`isSuccess`	`Boolean`	요청 성공 여부
`status`	`Number`	응답 상태
`code`	`Number`	응답 코드
`message`	`String`	응답 메시지
`timestamp`	`String`	api 호출 일시

게시글 수정 성공

Request Example

```
PATCH /posts/1 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Content-Length: 93
Host: mylog.cf

{
  "seriesId" : 6,
  "title" : "수정된 게시글",
  "content" : "수정됐어요!"
}
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 173

{
  "isSuccess" : true,
  "status" : 204,
  "code" : 201,
  "message" : "게시글 수정에 성공하였습니다.",
  "timestamp" : "2021-11-04T21:32:45.932+00:00"
}
```

3-4. 게시글 삭제

게시글 삭제

Request Header

Table 4. /posts/{id}

Parameter	Description
`id`	삭제 요청할 게시글 ID

Path	Type	Description
`isSuccess`	`Boolean`	요청 성공 여부
`status`	`Number`	응답 상태
`code`	`Number`	응답 코드
`message`	`String`	응답 메시지
`timestamp`	`String`	api 호출 일시

게시글 삭제 성공

Request Example

```
DELETE /posts/1 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Host: mylog.cf
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 173

{
  "isSuccess" : true,
  "status" : 204,
  "code" : 202,
  "message" : "게시글 삭제에 성공하였습니다.",
  "timestamp" : "2021-11-04T21:32:45.834+00:00"
}
```

---

4. Comment API

4-1. 댓글 등록

댓글 등록

Request Header

Path	Type	Description
`postId`	`Number`	게시글 번호
`writer`	`String`	작성자 닉네임 (없을 경우 익명)
`content`	`String`	댓글 내용
`password`	`String`	댓글 비밀번호

Path	Type	Description

isSuccess	Boolean	요청 성공 여부
status	Number	응답 상태
code	Number	응답 코드
message	String	응답 메시지
timestamp	String	api 호출 일시

댓글 등록 성공

Request Example

```
POST /comments HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Content-Length: 106
Host: mylog.cf

{
  "postId" : 1,
  "writer" : "홍길동",
  "password" : "1234",
  "content" : "새로운 댓글"
}
```

Response Example

```
HTTP/1.1 201 Created
Content-Type: application/json;charset=UTF-8
Content-Length: 170

{
  "isSuccess" : true,
  "status" : 201,
  "code" : 201,
  "message" : "댓글 등록에 성공하였습니다.",
  "timestamp" : "2021-11-04T21:32:45.278+00:00"
}
```

4-2. 댓글 조회

댓글 조회

Request Header

Path	Type	Description
isSuccess	Boolean	요청 성공 여부
status	Number	응답 상태
code	Number	응답 코드
message	String	응답 메시지
result.commentId	Number	댓글 번호
result.postId	Number	게시글 번호
result.commentWriter	String	댓글 작성자
result.commentContent	String	댓글 내용
result.commentCreatedAt	String	댓글 작성일
result.commentUpdatedAt	String	댓글 수정일
timestamp	String	api 호출 일시

댓글 조회 성공

Request Example

```
GET /comments/5 HTTP/1.1
```

```
Content-Type: application/json;charset=UTF-8
Accept: application/json
Host: mylog.cf
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 420

{
  "isSuccess" : true,
  "status" : 200,
  "code" : 201,
  "message" : "댓글 조회에 성공하였습니다.",
  "result" : {
    "commentId" : 5,
    "postId" : 1,
    "commentWriter" : "민정",
    "commentContent" : "역시 도커가 짱이에요",
    "commentCreatedAt" : "2021-11-03T23:01:40",
    "commentUpdatedAt" : "2021-11-05T02:53:22"
  },
  "timestamp" : "2021-11-04T21:32:45.360+00:00"
}
```

댓글 리스트 조회

Request Header

Parameter	Description
`postId`	게시글 번호
`page`	페이지 번호
`size`	페이지 사이즈

Path	Type	Description
`isSuccess`	`Boolean`	요청 성공 여부
`status`	`Number`	응답 상태
`code`	`Number`	응답 코드
`message`	`String`	응답 메시지
`page.currentPage`	`Number`	현재 페이지 번호
`page.pageSize`	`Number`	페이지 사이즈
`page.totalPages`	`Number`	전체 페이지 수
`page.totalElements`	`Number`	전체 요소 수
`result`	`Array`	댓글 조회 결과리스트
`result.[].commentId`	`Number`	댓글 번호
`result.[].postId`	`Number`	게시글 번호
`result.[].commentWriter`	`String`	댓글 작성자
`result.[].commentContent`	`String`	댓글 내용
`result.[].commentCreatedAt`	`String`	댓글 작성일
`result.[].commentUpdatedAt`	`String`	댓글 수정일
`timestamp`	`String`	api 호출 일시

댓글 리스트 조회 성공

Request Example

```
GET /comments/?postId=1&size=10&page=1 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Host: mylog.cf
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 538

{
  "isSuccess" : true,
  "status" : 200,
  "code" : 201,
  "message" : "댓글 조회에 성공하였습니다.",
  "page" : {
    "currentPage" : 0,
    "pageSize" : 10,
    "totalPages" : 1,
    "totalElements" : 1
  },
  "result" : [ {
    "commentId" : 5,
    "postId" : 1,
    "commentWriter" : "민정",
    "commentContent" : "역시 도커가 짱이에요",
    "commentCreatedAt" : "2021-11-03T23:01:40",
    "commentUpdatedAt" : "2021-11-05T02:53:22"
  } ],
  "timestamp" : "2021-11-04T21:32:45.700+00:00"
}
```

4-3. 댓글 삭제

댓글 삭제

Request Header

Table 5. /comments/{id}

Parameter	Description
`id`	삭제 요청할 댓글 ID

Path	Type	Description
`isSuccess`	`Boolean`	요청 성공 여부
`status`	`Number`	응답 상태
`code`	`Number`	응답 코드
`message`	`String`	응답 메시지
`timestamp`	`String`	api 호출 일시

댓글 삭제 성공

Request Example

```
DELETE /comments/5 HTTP/1.1
Content-Type: application/json;charset=UTF-8
Accept: application/json
Host: mylog.cf
```

Response Example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 170

{
  "isSuccess" : true,
  "status" : 204,
  "code" : 202,
```

```
  "message" : "댓글 삭제에 성공하였습니다.",
  "timestamp" : "2021-11-04T21:32:44.533+00:00"
}
```

Version 1.0
Last updated 2021-11-05 06:32:26 +0900