

# 인터페이스 설계서

---

## 1. 사용자 편의기능 지원 (SIR-001)

### 호출 방식

- **REST API**
- **Method:** GET, POST

### 요청 파라미터

### 사용자 도움말 조회

- **Endpoint:** `/api/help`
- **Method:** GET
- **Parameters:**
  - `page` (optional): 페이지 번호
  - `category` (optional): 도움말 카테고리

### 오류 메시지 조회

- **Endpoint:** `/api/errors`
- **Method:** GET
- **Parameters:**
  - `errorCode` (required): 오류 코드

### 응답 구조

### 사용자 도움말 조회 응답

```
{
  "status": "success",
  "data": {
    "currentPage": 1,
    "totalPages": 5,
    "helpItems": [
      {
        "id": "HLP001",
        "title": "로그인 방법",
        "content": "시스템에 로그인하는 방법에 대한 설명입니다."
      },
      {
        "id": "HLP002",
        "title": "비밀번호 재설정",
        "content": "비밀번호를 재설정하는 절차에 대한 안내입니다."
      }
    ]
  }
}
```

```
}  
}
```

#### 오류 메시지 조회 응답

```
{  
  "status": "success",  
  "data": {  
    "errorCode": "ERR404",  
    "message": "페이지를 찾을 수 없습니다."  
  }  
}
```

## 2. 내부 시스템 연계 (SIR-002)

#### 호출 방식

- **API 연동**
- **Method:** POST

#### 요청 파라미터

#### 내부 시스템 데이터 연계

- **Endpoint:** `/api/internal/sync`
- **Method:** POST
- **Parameters:**
  - `systemId` (required): 연계 대상 시스템 ID
  - `dataPayload` (required): 연계할 데이터의 JSON 객체

#### 응답 구조

```
{  
  "status": "success",  
  "message": "데이터 연계가 성공적으로 완료되었습니다.",  
  "timestamp": "2024-08-01T12:34:56Z"  
}
```

## 3. 모바일 웹 반응형 웹 (SIR-003)

#### 호출 방식

- **웹 애플리케이션**
- **Method:** N/A (웹사이트 접속)

#### 요청 파라미터

- 없음

## 응답 구조

- **HTML/CSS/JavaScript** 기반의 반응형 웹 인터페이스 제공

## 4. 커스터마이징 용이성 (SIR-004)

### 호출 방식

- 관리자 대시보드
- **Method:** N/A (웹 인터페이스를 통한 설정)

### 요청 파라미터

### 메뉴 구성 변경

- **Endpoint:** `/admin/menu/update`
- **Method:** POST
- **Parameters:**
  - `menuId` (required): 변경할 메뉴 ID
  - `newConfig` (required): 새로운 메뉴 구성 정보

### 응답 구조

```
{
  "status": "success",
  "message": "메뉴 구성이 성공적으로 업데이트되었습니다."
}
```

## 5. 사용자 친화 지원 (SIR-005)

### 호출 방식

- **REST API**
- **Method:** GET, POST

### 요청 파라미터

### 사용자 경험 데이터 조회

- **Endpoint:** `/api/user/experience`
- **Method:** GET
- **Parameters:**
  - `userId` (required): 사용자 ID

### 사용자 피드백 제출

- **Endpoint:** `/api/user/feedback`
- **Method:** POST
- **Parameters:**
  - `userId` (required): 사용자 ID
  - `feedback` (required): 피드백 내용

응답 구조

사용자 경험 데이터 조회 응답

```
{
  "status": "success",
  "data": {
    "userId": "USR123",
    "experienceScore": 4.5,
    "feedbacks": [
      {
        "date": "2024-07-20",
        "comment": "인터페이스가 직관적입니다."
      }
    ]
  }
}
```

사용자 피드백 제출 응답

```
{
  "status": "success",
  "message": "피드백이 성공적으로 제출되었습니다."
}
```

## 6. 요구사항 분석 및 제안 (SIR-006)

호출 방식

- **REST API**
- **Method:** POST

요청 파라미터

요구사항 분석 요청

- **Endpoint:** `/api/requirements/analyze`
- **Method:** POST
- **Parameters:**
  - `requirementId` (required): 분석할 요구사항 ID

- **details** (required): 요구사항 상세 내용

#### 응답 구조

```
{
  "status": "success",
  "message": "요구사항 분석이 완료되었습니다.",
  "analysisReportUrl": "https://example.com/reports/analysis123.pdf"
}
```

## 7. 각종 솔루션 연계 (SIR-007)

#### 호출 방식

- **API 연동**
- **Method:** POST

#### 요청 파라미터

#### 솔루션 연계 요청

- **Endpoint:** `/api/solutions/integrate`
- **Method:** POST
- **Parameters:**
  - **solutionId** (required): 연계할 솔루션 ID
  - **integrationDetails** (required): 연계 상세 정보

#### 응답 구조

```
{
  "status": "success",
  "message": "솔루션 연계가 성공적으로 완료되었습니다."
}
```

## 8. 시스템 간 연계 정의 (SIR-008)

#### 호출 방식

- **API 연동**
- **Method:** POST

#### 요청 파라미터

#### 시스템 간 데이터 연계 정의

- **Endpoint:** `/api/system/integrate`
- **Method:** POST

- **Parameters:**

- **sourceSystemId** (required): 원본 시스템 ID
- **targetSystemId** (required): 대상 시스템 ID
- **dataMapping** (required): 데이터 매핑 정보

#### 응답 구조

```
{
  "status": "success",
  "message": "시스템 간 연계 정의가 완료되었습니다."
}
```

## 9. 스포츠지원포털 통합회원 적용 및 SSO 적용 (SIR-009)

#### 호출 방식

- **SSO 연동**
- **Method:** GET, POST

#### 요청 파라미터

##### SSO 로그인 요청

- **Endpoint:** `/auth/sso/login`
- **Method:** POST
- **Parameters:**
  - **token** (required): SSO 토큰

##### 통합회원 정보 조회

- **Endpoint:** `/api/members/integrate`
- **Method:** GET
- **Parameters:**
  - **memberId** (required): 회원 ID

#### 응답 구조

##### SSO 로그인 응답

```
{
  "status": "success",
  "message": "SSO 로그인이 성공적으로 완료되었습니다.",
  "sessionToken": "SESSION12345"
}
```

##### 통합회원 정보 조회 응답

```
{
  "status": "success",
  "data": {
    "memberId": "MBR789",
    "name": "홍길동",
    "email": "hong@example.com",
    "joinedDate": "2022-01-15"
  }
}
```

## 10. 교육과정 정보 연계 (SIR-010)

### 호출 방식

- **REST API**
- **Method:** GET

### 요청 파라미터

### 교육과정 정보 조회

- **Endpoint:** `/api/courses`
- **Method:** GET
- **Parameters:**
  - `courseId` (optional): 교육과정 ID
  - `category` (optional): 교육 카테고리
  - `status` (optional): 교육 상태 (예: 진행중, 완료)

### 응답 구조

```
{
  "status": "success",
  "data": {
    "courses": [
      {
        "courseId": "CRS101",
        "title": "스포츠 심판 교육",
        "description": "스포츠 심판을 위한 기본 교육 과정입니다.",
        "startDate": "2024-09-01",
        "endDate": "2024-12-01",
        "status": "진행중"
      }
    ]
  }
}
```

## 11. 교육수료정보 연계 (SIR-011)

## 호출 방식

- **REST API**
- **Method:** POST

## 요청 파라미터

### 교육수료정보 전송

- **Endpoint:** `/api/completion/submit`
- **Method:** POST
- **Parameters:**
  - `userId` (required): 사용자 ID
  - `courseId` (required): 교육과정 ID
  - `completionStatus` (required): 수료 상태 (예: 수료, 미수료)
  - `completionDate` (optional): 수료 날짜

## 응답 구조

```
{
  "status": "success",
  "message": "교육수료정보가 성공적으로 연계되었습니다."
}
```

## 12. 교육수료증 신청정보 연계 (SIR-012)

## 호출 방식

- **REST API**
- **Method:** POST

## 요청 파라미터

### 교육수료증 신청

- **Endpoint:** `/api/certificate/apply`
- **Method:** POST
- **Parameters:**
  - `userId` (required): 사용자 ID
  - `courseId` (required): 교육과정 ID
  - `purpose` (required): 발급 용도
  - `quantity` (optional): 발급 부수

## 응답 구조

```
{
  "status": "success",
```



```
"message": "교육수료증 신청이 접수되었습니다.",  
"certificateId": "CRT456789"  
}
```

### 13. 교육수료증 발급이력정보 연계 (SIR-013)

호출 방식

- **REST API**
- **Method:** GET

요청 파라미터

교육수료증 발급이력 조회

- **Endpoint:** `/api/certificate/history`
- **Method:** GET
- **Parameters:**
  - `certificateId` (required): 수료증 ID

응답 구조

```
{  
  "status": "success",  
  "data": {  
    "certificateId": "CRT456789",  
    "userId": "USR123",  
    "courseId": "CRS101",  
    "issueDate": "2024-12-15",  
    "issuedBy": "AdminUser",  
    "status": "발급완료"  
  }  
}
```

### 14. API 호출 예시

사용자 피드백 제출 요청 예시

**Request:**

```
POST /api/user/feedback  
Content-Type: application/json  
  
{  
  "userId": "USR123",  
  "feedback": "인터페이스가 직관적입니다."  
}
```

**Response:**

```
{
  "status": "success",
  "message": "피드백이 성공적으로 제출되었습니다."
}
```

## 교육수료증 신청 요청 예시

**Request:**

```
POST /api/certificate/apply
Content-Type: application/json
```

```
{
  "userId": "USR123",
  "courseId": "CRS101",
  "purpose": "이력서 첨부",
  "quantity": 1
}
```

**Response:**

```
{
  "status": "success",
  "message": "교육수료증 신청이 접수되었습니다.",
  "certificateId": "CRT456789"
}
```

## 15. 에러 처리

## 공통 에러 응답 구조

```
{
  "status": "error",
  "error": {
    "code": "ERR_CODE",
    "message": "에러 메시지"
  }
}
```

## 예시

### 잘못된 요청 파라미터

```
{
  "status": "error",
  "error": {
    "code": "ERR400",
    "message": "잘못된 요청 파라미터입니다."
  }
}
```

### 인증 실패

```
{
  "status": "error",
  "error": {
    "code": "ERR401",
    "message": "인증에 실패했습니다."
  }
}
```

### 서버 오류

```
{
  "status": "error",
  "error": {
    "code": "ERR500",
    "message": "서버 오류가 발생했습니다."
  }
}
```

## 16. 보안 고려사항

- **인증 및 권한 부여:** 모든 API 호출은 유효한 인증 토큰을 요구합니다.
- **데이터 암호화:** 민감한 데이터는 전송 시 TLS를 사용하여 암호화합니다.
- **입력 검증:** 모든 입력 데이터는 서버 측에서 철저히 검증하여 SQL Injection, XSS 등의 공격을 방지합니다.
- **로그 기록:** 모든 API 호출은 보안 로그에 기록되며, 접근 기록은 최소 3개월 동안 보관됩니다.

## 17. 데이터 형식 및 표준

- **데이터 형식:** 모든 요청 및 응답은 JSON 형식을 사용합니다.
- **날짜 형식:** ISO 8601 형식 (YYYY-MM-DDTHH:MM:SSZ)
- **코딩 표준:** 전자정부 표준 프레임워크 준수

이 문서는 RFP에서 요구된 인터페이스 요구사항을 분석하여 작성되었습니다. 상세한 내용은 각 인터페이스의 요구사항에 따라 추가 및 수정될 수 있습니다.