

자치행정과 AI Agent [JJ-101-301] 개발 방안

1. AI Agent 개발 방법 및 절차

1.1 자치행정과 업무 범위 및 기능 정의

주요 업무 영역

- 행정구역 관리: 읍면동 경계 조정, 행정구역 변경 업무
- 지방자치제도 운영: 자치법규 제정·개정, 조례 및 규칙 관리
- 선거 관련 업무: 지방선거 지원, 선거구 관리
- 주민자치 활성화: 주민자치회 운영 지원, 주민참여예산제
- 행정절차 관리: 행정절차법 준수, 민원처리 절차 표준화
- 법제 업무: 자치법규 심사, 법령 해석 및 자문
- 감사 및 평가: 행정감사, 성과평가 업무

AI Agent 핵심 기능

1. 조례·규칙 자동 검색 및 해석 시스템
2. 주민자치 프로그램 매칭 및 안내
3. 행정절차 가이드 및 진행상황 추적
4. 선거 관련 정보 제공 및 안내
5. 법령 해석 및 행정 자문

1.2 시스템 아키텍처

- 자치행정과 AI Agent [JJ-101-301]
 - — 조례·규칙 관리 모듈
 - — 자치법규 검색 엔진
 - — 조례 제정 프로세스 관리
 - — 법령 해석 AI
 - — 주민자치 지원 모듈
 - — 주민자치회 운영 지원
 - — 주민참여예산 관리
 - — 지역 커뮤니티 매칭
 - — 선거 지원 모듈
 - — 선거 정보 제공
 - — 선거구 조회 시스템
 - — 후보자 정보 관리
 - — 행정절차 관리 모듈
 - — 민원처리 절차 안내
 - — 행정감사 지원
 - — 성과평가 시스템

1.3 개발 절차

Phase 1: 요구사항 분석 및 설계 (1주)

- 자치행정과 업무 프로세스 분석
- 기존 시스템 연동 방안 설계
- 사용자 시나리오 정의
- 데이터 모델 설계

Phase 2: 데이터 수집 및 전처리 (2주)

- 공개 데이터 수집 및 정제
- 비공개 데이터 접근 권한 확보
- 데이터 라벨링 및 검증
- 훈련 데이터셋 구축

Phase 3: AI 모델 개발 (3주)

- DeepSeek R1 기반 모델 Fine-tuning
- 도메인 특화 어댑터 개발
- 성능 최적화 및 검증
- 테스트 데이터셋 평가

Phase 4: 시스템 통합 및 테스트 (2주)

- 포털 AI Agent와 연동
- 기존 행정시스템 API 연결
- 통합 테스트 및 성능 검증
- 보안 테스트 및 취약점 점검

2. DeepSeek R1 Fine-tuning 데이터셋

2.1 공개 데이터

2.1.1 법령 및 조례 데이터

출처: 국가법령정보센터, 자치법규정보시스템

- 데이터 규모: 100,000+ 조례/규칙 문서
- 포함 내용:
 - 제주특별자치도 조례 전문 (2,500건)
 - 전국 지방자치단체 조례 (95,000건)
 - 법령 해석 사례 (3,000건)

- 전처리 방법:

조례 텍스트 전처리 예시 `def preprocess_ordinance(text):` # 조문 번호 정규화

```
text = re.sub(r'제\d+조', '[조문]', text) # 법령 인용 정규화
text = re.sub(r'「[^」」+', '[법령명]', text) return text
```

-

2.1.2 행정절차 가이드 데이터

출처: 정부24, 각 지자체 홈페이지

- 데이터 규모: 50,000+ 절차 문서
- 포함 내용:
 - 민원처리 매뉴얼 (15,000건)
 - 행정절차 Q&A (20,000건)
 - 업무처리 가이드라인 (15,000건)

2.1.3 선거 관련 공개 데이터

출처: 중앙선거관리위원회

- 데이터 규모: 30,000+ 선거 관련 문서
- 포함 내용:
 - 선거 FAQ (5,000건)
 - 선거법 해석 사례 (10,000건)
 - 선거구 정보 (15,000건)

2.2 비공개 데이터

2.2.1 내부 업무 문서

출처: 제주특별자치도 자치행정과

- 데이터 규모: 20,000+ 내부 문서
- 포함 내용:
 - 조례 제정 과정 문서 (3,000건)
 - 내부 업무 매뉴얼 (2,000건)
 - 법령 해석 내부 자문 (5,000건)
 - 주민자치회 운영 사례 (10,000건)
- 보안 처리:
 - 개인정보 마스킹 처리
 - 민감 정보 제거
 - 접근 권한 관리

2.2.2 민원 처리 이력 데이터

출처: 제주특별자치도 민원시스템

- 데이터 규모: 100,000+ 민원 기록
- 포함 내용:
 - 자치행정과 관련 민원 (30,000건)
 - 민원 처리 과정 로그 (50,000건)
 - 민원인-담당자 대화 기록 (20,000건)

- 익명화 처리:

```
def anonymize_complaint_data(data): # 개인정보 마스킹
    data['name'] = hash_personal_info(data['name'])
    data['phone'] = mask_phone_number(data['phone'])
    data['address'] = generalize_address(data['address'])
    return data
```
-

2.2.3 주민자치 활동 데이터

출처: 제주특별자치도 주민자치과

- 데이터 규모: 15,000+ 활동 기록
- 포함 내용:
 - 주민자치회 회의록 (5,000건)
 - 주민참여예산 제안서 (8,000건)
 - 지역 커뮤니티 활동 로그 (2,000건)

2.3 Fine-tuning 데이터셋 구성

훈련 데이터 분할

- 훈련용 (70%): 231,000건
- 검증용 (15%): 49,500건
- 테스트용 (15%): 49,500건

도메인별 가중치

- domain_weights:
- legal_interpretation: 0.3
- administrative_procedure: 0.25
- resident_autonomy: 0.2
- election_support: 0.15
- general_admin: 0.1

3. 개발자 일일 일정표 (시연용 시스템)

3.1 개발 기간: 8주 (40일)

1주차: 프로젝트 설정 및 기반 구축

1일차 (월요일)

오전 (09:00-13:00)

- 프로젝트 환경 설정 (Docker, Kubernetes)
- GitHub 리포지토리 생성 및 CI/CD 파이프라인 구축

- DeepSeek R1 모델 다운로드 및 환경 테스트
- 개발 도구 및 라이브러리 설치

오후 (14:00-18:00)

- 자치행정과 업무 프로세스 문서 분석
- 시스템 요구사항 정의서 작성
- API 설계 문서 초안 작성
- 데이터베이스 스키마 설계

2일차 (화요일)

오전 (09:00-13:00)

- 공개 데이터 수집 스크립트 개발 (조례, 법령 데이터)
- 국가법령정보센터 API 연동
- 자치법규정보시스템 크롤링 스크립트 작성
- 데이터 수집 자동화 파이프라인 구축

오후 (14:00-18:00)

- 수집된 데이터 전처리 모듈 개발
- 텍스트 정규화 및 토큰화 함수 구현
- 데이터 품질 검증 도구 개발
- 초기 데이터셋 검토 및 분석

3일차 (수요일)

오전 (09:00-13:00)

- FastAPI 기반 백엔드 서버 구축
- 기본 REST API 엔드포인트 구현
- 데이터베이스 연결 및 ORM 설정
- 로깅 및 모니터링 시스템 구축

오후 (14:00-18:00)

- React 기반 프론트엔드 초기 설정
- 컴포넌트 구조 설계 및 라우팅 설정
- UI/UX 프로토타입 개발
- 기본 대화형 인터페이스 구현

4일차 (목요일)

오전 (09:00-13:00)

- DeepSeek R1 모델 로딩 및 추론 테스트
- 모델 서빙 환경 구축 (vLLM 설정)
- GPU 메모리 최적화 설정
- 기본 프롬프트 템플릿 작성

오후 (14:00-18:00)

- 조례 검색 엔진 프로토타입 개발
- Elasticsearch 설정 및 인덱싱
- 키워드 기반 검색 기능 구현
- 검색 결과 랭킹 알고리즘 구현

5일차 (금요일)

오전 (09:00-13:00)

- 1주차 개발 결과 통합 테스트
- 시스템 아키텍처 검토 및 개선
- 코드 리뷰 및 리팩토링
- 다음 주 계획 수립

오후 (14:00-18:00)

- 문서화 작업 (API 문서, 설치 가이드)
- 개발 환경 백업 및 버전 관리
- 주간 진행 상황 보고서 작성
- 기술적 이슈 분석 및 해결 방안 모색

2주차: 데이터 처리 및 AI 모델 통합

6일차 (월요일)

오전 (09:00-13:00)

- 행정절차 데이터 수집 모듈 개발
- 정부24 API 연동 및 데이터 추출
- 민원처리 매뉴얼 파싱 로직 구현
- 데이터 품질 향상을 위한 전처리 개선

오후 (14:00-18:00)

- 자연어 처리 파이프라인 구축
- 한국어 형태소 분석기 통합 (KoNLPy)
- 문서 분류 및 개체명 인식 모델 적용
- 텍스트 임베딩 생성 시스템 구축

7일차 (화요일)

오전 (09:00-13:00)

- DeepSeek R1 Fine-tuning 환경 설정
- 훈련 데이터셋 전처리 및 포맷 변환
- Lora 어댑터 설정 및 하이퍼파라미터 튜닝
- 분산 학습 환경 구축

오후 (14:00-18:00)

- 모델 Fine-tuning 시작 및 모니터링
- 훈련 과정 로그 분석 및 최적화
- 검증 데이터를 이용한 성능 평가
- 오버피팅 방지를 위한 조기 종료 설정

8일차 (수요일)

오전 (09:00-13:00)

- 조례 해석 AI 모듈 개발
- 법령 조문 분석 및 해석 로직 구현
- 유사 조례 검색 및 비교 기능 개발
- 법령 용어 사전 구축 및 통합

오후 (14:00-18:00)

- 주민자치 지원 모듈 개발
- 주민자치회 활동 매칭 알고리즘 구현
- 지역별 커뮤니티 정보 관리 시스템 구축
- 참여 프로그램 추천 엔진 개발

9일차 (목요일)

오전 (09:00-13:00)

- 선거 지원 시스템 모듈 개발
- 선거구 조회 API 구현
- 선거 일정 및 정보 제공 시스템 구축
- 후보자 정보 관리 데이터베이스 설계

오후 (14:00-18:00)

- 행정절차 안내 시스템 개발
- 단계별 절차 가이드 생성 로직 구현
- 진행상황 추적 시스템 구축
- 필요 서류 자동 생성 기능 개발

10일차 (금요일)

오전 (09:00-13:00)

- 2주차 개발 결과 통합 및 테스트
- AI 모델 성능 평가 및 최적화
- 각 모듈 간 연동 테스트
- 시스템 안정성 검증

오후 (14:00-18:00)

- 사용자 인터페이스 개선
- 대화형 AI 응답 품질 향상
- 에러 처리 및 예외 상황 대응 로직 구현
- 주간 진행 상황 정리 및 문서화

3주차: 시스템 통합 및 최적화

11일차 (월요일)

오전 (09:00-13:00)

- 포털 AI Agent와의 연동 인터페이스 개발
- API Gateway 설정 및 라우팅 로직 구현
- 인증 및 권한 관리 시스템 통합
- 메시지 큐를 이용한 비동기 처리 구현

오후 (14:00-18:00)

- 실시간 대화 처리 시스템 최적화
- WebSocket 기반 실시간 통신 구현
- 세션 관리 및 대화 컨텍스트 유지 로직
- 다중 사용자 동시 처리 능력 향상

12일차 (화요일)

오전 (09:00-13:00)

- 데이터베이스 성능 최적화
- 인덱싱 전략 수립 및 적용
- 쿼리 최적화 및 실행 계획 분석
- 캐싱 시스템 도입 (Redis)

오후 (14:00-18:00)

- AI 모델 추론 성능 최적화
- 모델 양자화 및 가속화 적용
- 배치 처리를 통한 처리량 향상
- GPU 메모리 사용량 최적화

13일차 (수요일)

오전 (09:00-13:00)

- 보안 시스템 강화
- HTTPS 및 API 보안 설정
- 입력 데이터 검증 및 SQL 인젝션 방지
- 사용자 데이터 암호화 적용

오후 (14:00-18:00)

- 모니터링 및 로깅 시스템 구축
- Prometheus, Grafana를 이용한 메트릭 수집
- 에러 추적 및 알림 시스템 구축
- 성능 모니터링 대시보드 개발

14일차 (목요일)

오전 (09:00-13:00)

- 통합 테스트 시나리오 작성 및 실행
- 각 모듈별 단위 테스트 보완
- 부하 테스트 및 성능 벤치마킹
- 메모리 누수 및 리소스 사용량 분석

오후 (14:00-18:00)

- 사용자 경험 개선
- UI/UX 피드백 반영 및 개선
- 접근성 향상 (웹 접근성 가이드라인 준수)
- 모바일 반응형 디자인 적용

15일차 (금요일)

오전 (09:00-13:00)

- 3주차 전체 시스템 통합 테스트
- 엔드투엔드 테스트 시나리오 실행
- 성능 및 안정성 최종 검증
- 발견된 버그 수정 및 개선

오후 (14:00-18:00)

- 배포 준비 작업
- Docker 이미지 최적화 및 빌드
- Kubernetes 배포 설정 파일 작성
- CI/CD 파이프라인 최종 점검

4주차: 고급 기능 구현

16일차 (월요일)

오전 (09:00-13:00)

- 다국어 지원 시스템 구축
- 한국어, 영어, 중국어, 일본어 번역 모듈 통합
- 언어별 UI 현지화 작업
- 다국어 모델 성능 최적화

오후 (14:00-18:00)

- 음성 인터페이스 개발
- STT(Speech-to-Text) 엔진 통합
- TTS(Text-to-Speech) 엔진 통합
- 음성 명령 처리 로직 구현

17일차 (화요일)

오전 (09:00-13:00)

- 고급 검색 기능 구현
- 의미 기반 검색 (Semantic Search) 적용
- 검색 결과 개인화 알고리즘 구현
- 검색 히스토리 및 추천 시스템 개발

오후 (14:00-18:00)

- 예측 분석 시스템 개발
- 민원 패턴 분석 및 예측 모델 구현
- 업무량 예측 및 리소스 할당 최적화
- 트렌드 분석 대시보드 개발

18일차 (수요일)

오전 (09:00-13:00)

- 워크플로우 자동화 시스템 구현
- 복잡한 행정절차의 자동화 로직 개발
- 단계별 승인 시스템 구축
- 자동 알림 및 리마인더 시스템 개발

오후 (14:00-18:00)

- 데이터 분석 및 리포팅 시스템
- 업무 처리 통계 및 분석 기능 구현
- 성과 지표 추적 시스템 개발
- 자동 보고서 생성 기능 구현

19일차 (목요일)

오전 (09:00-13:00)

- 기계학습 기반 개선 시스템
- 사용자 피드백 학습 모델 구현
- A/B 테스트 프레임워크 구축
- 지속적 학습 파이프라인 개발

오후 (14:00-18:00)

- 고급 보안 기능 구현
- 다단계 인증 시스템 구축

- 블록체인 기반 무결성 검증 시스템
- 개인정보 자동 마스킹 시스템 개발

20일차 (금요일)

오전 (09:00-13:00)

- 4주차 고급 기능 통합 테스트
- 전체 시스템 성능 재평가
- 새로운 기능들의 안정성 검증
- 사용자 수용성 테스트 준비

오후 (14:00-18:00)

- 기술 문서 작성 및 업데이트
- API 문서 최신화
- 사용자 매뉴얼 작성
- 시스템 아키텍처 문서 정리

5주차: 시연 준비 및 최적화

21일차 (월요일)

오전 (09:00-13:00)

- 시연용 데이터 준비 및 시나리오 작성
- 실제 업무 상황을 모방한 테스트 케이스 개발
- 시연 스크립트 작성 및 연습
- 데모 환경 설정 및 최적화

오후 (14:00-18:00)

- 사용자 인터페이스 최종 개선
- 시각적 효과 및 애니메이션 추가
- 응답 속도 최적화
- 오류 메시지 개선 및 사용자 가이드 강화

22일차 (화요일)

오전 (09:00-13:00)

- 실시간 대시보드 개발
- 시스템 상태 실시간 모니터링
- 사용 통계 실시간 표시
- 성능 지표 시각화

오후 (14:00-18:00)

- 장애 복구 시스템 구축
- 자동 페일오버 메커니즘 구현

- 백업 및 복구 절차 자동화
- 재해 복구 테스트 실행

23일차 (수요일)

오전 (09:00-13:00)

- 성능 튜닝 및 최적화
- 데이터베이스 쿼리 최적화
- 캐시 전략 재검토 및 개선
- 메모리 사용량 최적화

오후 (14:00-18:00)

- 보안 최종 점검
- 보안 취약점 스캔 및 수정
- 침투 테스트 시나리오 실행
- 보안 정책 문서화

24일차 (목요일)

오전 (09:00-13:00)

- 사용자 교육 자료 준비
- 사용자 가이드 비디오 제작
- FAQ 문서 작성
- 교육 시나리오 개발

오후 (14:00-18:00)

- 시스템 배포 준비
- 프로덕션 환경 설정
- 배포 스크립트 작성 및 테스트
- 롤백 계획 수립

25일차 (금요일)

오전 (09:00-13:00)

- 5주차 전체 시스템 최종 테스트
- 시연 시나리오 리허설
- 성능 벤치마크 최종 측정
- 예상 질문에 대한 답변 준비

오후 (14:00-18:00)

- 프로젝트 문서 최종 정리
- 코드 주석 및 문서화 완성
- 향후 개발 로드맵 작성
- 유지보수 가이드 문서 작성

6주차: 고도화 및 확장성 개선

26일차 (월요일)

오전 (09:00-13:00)

- 마이크로서비스 아키텍처 적용
- 모놀리식 구조를 마이크로서비스로 분리
- 서비스 간 통신 프로토콜 구현
- API Gateway 고도화

오후 (14:00-18:00)

- 컨테이너 오케스트레이션 최적화
- Kubernetes 설정 고도화
- 자동 스케일링 정책 수립
- 리소스 사용량 최적화

27일차 (화요일)

오전 (09:00-13:00)

- 데이터 파이프라인 구축
- 실시간 데이터 스트리밍 시스템 구현
- ETL 프로세스 자동화
- 데이터 품질 모니터링 시스템

오후 (14:00-18:00)

- 고가용성 시스템 구축
- 다중 데이터센터 배포 전략
- 로드 밸런싱 최적화
- 장애 감지 및 자동 복구 시스템

28일차 (수요일)

오전 (09:00-13:00)

- AI 모델 버전 관리 시스템
- 모델 배포 자동화 파이프라인
- A/B 테스트를 통한 모델 성능 비교
- 모델 롤백 메커니즘 구현

오후 (14:00-18:00)

- 빅데이터 처리 시스템 구축
- Apache Spark 클러스터 설정
- 대용량 데이터 배치 처리 최적화
- 실시간 분석 시스템 구현

29일차 (목요일)

오전 (09:00-13:00)

- 지능형 캐싱 시스템 구현
- 예측 기반 프리로딩 시스템
- 캐시 무효화 전략 개선
- 분산 캐시 시스템 구축

오후 (14:00-18:00)

- 사용자 행동 분석 시스템
- 클릭스트림 데이터 수집 및 분석
- 사용자 여정 맵핑
- 개인화 추천 시스템 고도화

30일차 (금요일)

오전 (09:00-13:00)

- 6주차 고도화 작업 통합 테스트
- 시스템 확장성 테스트
- 대용량 트래픽 부하 테스트
- 성능 병목 지점 분석 및 개선

오후 (14:00-18:00)

- 운영 자동화 도구 개발
- ChatOps 구현 (Slack 연동)
- 자동 배포 및 모니터링 시스템
- 인시던트 대응 자동화

7주차: 검증 및 최종 조정

31일차 (월요일)

오전 (09:00-13:00)

- 전체 시스템 스트레스 테스트
- 동시 사용자 1000명 부하 테스트
- 메모리 및 CPU 사용량 분석
- 데이터베이스 커백션 풀 최적화

오후 (14:00-18:00)

- 보안 침투 테스트
- 외부 보안 전문가와 협력한 테스트
- 취약점 분석 및 보완
- 보안 정책 최종 검토

32일차 (화요일)

오전 (09:00-13:00)

- 사용성 테스트 및 개선
- 실제 사용자 그룹 테스트 진행
- 사용자 피드백 수집 및 분석
- UI/UX 최종 개선 사항 적용

오후 (14:00-18:00)

- 접근성 개선 작업
- 웹 접근성 가이드라인 준수 확인
- 스크린 리더 호환성 테스트
- 키보드 네비게이션 개선

33일차 (수요일)

오전 (09:00-13:00)

- 다국어 지원 최종 검증
- 번역 품질 검토 및 개선
- 문화적 차이 고려한 UI 조정
- 언어별 성능 최적화

오후 (14:00-18:00)

- 모바일 최적화 작업
- 모바일 전용 기능 개발
- 터치 인터페이스 개선
- 모바일 성능 최적화

34일차 (목요일)

오전 (09:00-13:00)

- 법적 컴플라이언스 검토
- 개인정보보호법 준수 확인
- 데이터 보존 정책 점검
- 법적 요구사항 대응 방안 수립

오후 (14:00-18:00)

- 최종 통합 테스트
- 모든 기능의 종합적 검증
- 예외 상황 처리 테스트
- 데이터 일관성 검증

35일차 (금요일)

오전 (09:00-13:00)

- 성능 메트릭 최종 측정
- 응답 시간, 처리량, 가용성 측정
- SLA 달성도 확인
- 성능 개선 여지 분석

오후 (14:00-18:00)

- 프로덕션 배포 최종 준비
- 배포 체크리스트 작성
- 모니터링 설정 최종 점검
- 백업 및 복구 절차 확인

8주차: 배포 및 시연

36일차 (월요일)

오전 (09:00-13:00)

- 프로덕션 환경 배포
- Blue-Green 배포 전략 적용
- 배포 후 헬스 체크 및 모니터링
- 성능 지표 실시간 확인

오후 (14:00-18:00)

- 시연 최종 준비
- 시연 시나리오 최종 점검
- 예상 질문 답변 준비
- 백업 계획 수립

37일차 (화요일)

오전 (09:00-13:00)

- 사용자 교육 및 온보딩
- 관리자 교육 프로그램 진행
- 사용자 매뉴얼 배포
- 지원 체계 구축

오후 (14:00-18:00)

- 모니터링 대시보드 최종 설정
- 알림 규칙 설정 및 테스트
- 로그 분석 시스템 구축
- 운영 매뉴얼 작성

38일차 (수요일)

오전 (09:00-13:00)

- 파일럿 테스트 진행
- 제한된 사용자 그룹 대상 테스트
- 실시간 피드백 수집
- 즉시 개선 사항 적용

오후 (14:00-18:00)

- 시연 리허설
- 전체 시연 시나리오 실행
- 예상 문제 상황 대응 연습
- 발표 자료 최종 점검

39일차 (목요일)

오전 (09:00-13:00)

- 최종 시스템 점검
- 모든 기능 정상 작동 확인
- 데이터 백업 상태 점검
- 보안 설정 최종 확인

오후 (14:00-18:00)

- 향후 발전 계획 수립
- 추가 기능 개발 로드맵 작성
- 유지보수 계획 수립
- 확장 가능성 분석

40일차 (금요일)

오전 (09:00-13:00)

- 공식 시연 진행
- 주요 기능 시연
- 성능 지표 발표
- 질의응답 세션

오후 (14:00-18:00)

- 시연 결과 정리
- 피드백 수집 및 분석
- 개선 계획 수립
- 프로젝트 완료 보고서 작성

4. 성과 지표 및 평가 기준

4.1 기술적 성과 지표

- 응답 시간: 평균 2초 이내
- 정확도: 조례 해석 정확도 95% 이상
- 가용성: 99.9% 이상
- 동시 사용자: 500명 이상 지원

4.2 사용자 만족도 지표

- 사용자 만족도: 4.5/5.0 이상
- 업무 효율성 향상: 70% 이상
- 민원 처리 시간 단축: 80% 이상

4.3 비즈니스 성과 지표

- 비용 절감: 연간 30% 이상
- 업무 자동화율: 85% 이상
- 민원 재접수율: 10% 이하

이 개발 계획을 통해 제주특별자치도 자치행정과의 업무를 효과적으로 지원하는 AI Agent를 구축할 수 있을 것으로 예상됩니다.

-