

AGENT LAND_SAARLAND Initialisierungs-Template

markdown

```
<template_init_agentland_saarland>
```

```
# AGENT LAND_SAARLAND Initialisierungsvorlage
```

Vorlagenparameter

- PROJECT_NAME: Name des Projekts (Standard: AGENT LAND_SAARLAND)
- PROJECT_DIR: Installationsverzeichnis (Standard: Aktuelles Verzeichnis)
- REGIONAL_CONFIG: Regionale Konfigurationsvorlage (Standard: saarland-standard)
- LANGUAGE_MODE: Primärer Sprachmodus (Standard: bilingual-de-en)

Verzeichnisstruktur erstellen

Erstelle folgende Verzeichnisstruktur:

AGENT LAND_SAARLAND/ |—— .claude/ # Claude-spezifisches Konfigurationsverzeichnis | |——
commands/ # Benutzerdefinierte Slash-Befehle | | |—— init.md # Initialisierungsbefehl | | |——
memory.md # Speicherverwaltungsbefehl | | |—— regional/ # Regionalspezifische Befehle | | |——
saarland.md # Saarland-spezifische Befehle | |—— config.json # Claude-Konfiguration | |—— memory/
Persistenter Speicher |—— .claudeignore # Von Claude zu ignorierende Dateien |—— .clauderules #
Regeln für Claudes Verhalten |—— CLAUDE.md # Projektspezifische Anweisungen für Claude |——
ai_docs/ # KI-Dokumentation | |—— models/ # Modelldokumentation | |—— prompts/ # Prompt-
Vorlagen und Richtlinien | |—— regional/ # Regionalspezifische Dokumentation | |—— saarland/ #
Saarland-spezifische Dokumentation |—— specs/ # Spezifikationen | |—— architecture/ #
Architekturdiagramme und -dokumente | |—— requirements/ # Anforderungsdokumentation | |——
regional/ # Regionalspezifische Spezifikationen | |—— saarland/ # Saarland-spezifische Spezifikationen
|—— src/ # Quellcode | |—— agent/ # Agent-Implementierung | |—— data/ # Datenressourcen | | |——
regional/ # Regionalspezifische Daten | | |—— saarland/ # Saarland-spezifische Daten | |—— tools/ #
Benutzerdefinierte Tool-Implementierung |—— config/ # Konfigurationsdateien | |—— regional/ #
Regionalspezifische Konfigurationen | |—— saarland.yaml # Saarland-spezifische Konfiguration |——
README.md # Projekt-README

Konfigurationsdateien

1. .claudeignore

Build-Artefakte

/build/ /dist/ /node_modules/ /**pycache/** *.pyc

Sensible Informationen

.env .env.* *.key _key.secrets/

Große Datendateien

*.csv *.db *.sqlite /data/raw/

Temporäre Dateien

.tmp/ .cache/ *.log *.bak

Systemdateien

.DS_Store Thumbs.db

2. .clauserules

```
```yaml
version: "1.0"

Allgemeine Regeln für Claudes Verhalten
general:
 assistantName: "AGENT_LAND_SAARLAND Assistent"
 defaultLanguage: "de"
 fallbackLanguage: "en"
 responseStyle: "professional"

Regionale Governance
regionalGovernance:
 region: "saarland"
 country: "germany"
 dataSovereignty: true
 complianceFrameworks:
 - "GDPR"
 - "German-AI-Act"
 - "SaarlandDigitalStrategy"

Erlaubte Aktionen
permissions:
 canModifyFiles:
 - "src/**/*.py"
 - "src/**/*.js"
 - "ai_docs/**/*.*"
 - "specs/**/*.*"
 canExecuteCommands: true

Richtlinien zur Datenverarbeitung
dataHandling:
 allowPII: false
 dataRetentionPolicy: "minimal"

Ethische Richtlinien
ethics:
 prioritizeSovereignty: true
 transparencyLevel: "high"
```

### 3. .claude/config.json

json

```
{
 "project": "AGENT_LAND_SAARLAND",
 "version": "1.0.0",
 "description": "KI-Projektvorlage für die Region Saarland",
 "defaultModel": "claude-3.5-sonnet",
 "regionalSettings": {
 "region": "saarland",
 "country": "germany",
 "language": "de-DE",
 "fallbackLanguage": "en-US"
 },
 "memory": {
 "enablePersistence": true,
 "storageLocation": ".claude/memory"
 },
 "security": {
 "dataGovernance": "eu-strict",
 "technicalSovereignty": "high",
 "complianceFrameworks": ["GDPR", "German-AI-Act"]
 }
}
```

#### 4. CLAUDE.md

markdown

## # AGENT\_LAND\_SAARLAND

### ## Projektübersicht

Dies ist eine KI-Projektvorlage für die Region Saarland in Deutschland, die sich auf

### ## Entwicklungsrichtlinien

- Der Quellcode sollte den PEP 8-Standards für Python und dem Airbnb-Styleguide für .
- Dokumentation sollte zweisprachig sein (Deutsch und Englisch)
- Die regionale Datenverarbeitung muss den DSGVO- und dem deutschen KI-Gesetz entspr

### ## Häufige Befehle

- Projekt initialisieren: ``/init``
- Speicher verwalten: ``/memory``
- Compliance überprüfen: ``/regional:saarland --compliance-check``

### ## Regionale Besonderheiten

- Technische Souveränität hat Priorität
- Daten müssen innerhalb deutscher Grenzen bleiben
- Regionale Dialektmerkmale sollten bei NLP-Aufgaben berücksichtigt werden
- Lokale Governance-Frameworks müssen respektiert werden

### ## Wichtige Dateien und Verzeichnisse

- Projektkonfiguration: ``config/regional/saarland.yaml``
- Regionale Datenquellen: ``src/data/regional/saarland/``
- Spezifikationen: ``specs/regional/saarland/``
- KI-Dokumentation: ``ai_docs/regional/saarland/``

## 5. config/regional/saarland.yaml

yaml

*# Regionale Konfiguration für das Saarland*

region:

name: "Saarland"

country: "Deutschland"

language:

primary: "de-DE"

secondary: "en-US"

dialect:

enable: true

name: "Saarländisch"

*# Einstellungen für technische Souveränität*

sovereignty:

level: "high"

dataSovereignty: true

dataLocalization: "germany-only"

complianceFrameworks:

- "DSGVO"
- "Deutsches KI-Gesetz"
- "Saarländische Digitalstrategie"

*# Regionale Datenquellen*

dataSources:

- name: "saarland-open-data"  
url: "https://www.saarland.de/opendata/api"  
type: "REST"  
description: "Saarland Open Data Portal"
- name: "saarland-geo"  
url: "https://geoportal.saarland.de/services"  
type: "WMS"  
description: "Geographische Daten für die Region Saarland"

*# Regionale API-Integrationen*

apiIntegrations:

- name: "dfki-ai-services"  
url: "https://dfki.de/services/ai"  
description: "KI-Dienste vom DFKI (Deutsches Forschungszentrum für Künstliche In"
- name: "saarland-university-nlp"  
url: "https://nlp.uni-saarland.de/api"  
description: "NLP-Dienste der Universität des Saarlandes"



## Benutzerdefinierte Befehle

### 1. .claude/commands/init.md

markdown

## # Projektinitialisierungsbefehl

Sie haben die Aufgabe, eine neue Instanz des AGENT LAND SAARLAND-Projekts zu initialisieren. Führen Sie diese Schritte sorgfältig aus:

1. Überprüfen Sie, ob das Projektverzeichnis leer ist oder nur die Vorlagendateien enthält
2. Erstellen Sie alle erforderlichen Verzeichnisse gemäß der Projektstruktur
3. Generieren Sie Konfigurationsdateien mit geeigneten Standardwerten für die Region
4. Validieren Sie, dass alle Komponenten korrekt installiert und zugänglich sind
5. Erstellen Sie einen Einrichtungsbericht, der den Initialisierungsprozess detailliert dokumentiert

Das Projekt sollte mit den folgenden saarlandspezifischen Einstellungen konfiguriert werden:

- Primärsprache: Deutsch
- Regionales Compliance-Framework: DSGVO + Deutsches KI-Gesetz + Saarländische Digitalisierungsverordnung
- Datenresidenz: Innerhalb deutscher Grenzen
- Technische Souveränität: Hohe Priorität
- Regionale API-Endpunkte: Saarland Open Data Portal

Berichten Sie über alle während der Initialisierung aufgetretenen Probleme und geben Sie die Lösungsmöglichkeiten an.



## 2. .claude/commands/memory.md

## # Speicherverwaltungsbefehl

Sie haben die Aufgabe, das Speichersystem für das AGENT\_LAND\_SAARLAND-Projekt zu verwalten. Dieser Befehl ermöglicht die Initialisierung, Abfrage und Verwaltung des persistenten Speichers.

### ## Parameter

- `--init`: Initialisieren eines neuen Speichers
- `--query <Begriff>`: Durchsuchen des Speichers nach bestimmten Informationen
- `--add <Datei>`: Hinzufügen von Inhalten aus einer Datei zum Speicher
- `--clear`: Löschen aller Inhalte aus dem Speicher
- `--backup`: Erstellen einer Sicherung des aktuellen Speichers
- `--restore <Sicherung>`: Wiederherstellen aus einer Sicherung

### ## Verwendungsbeispiele

1. Speicher initialisieren: `/memory --init`
2. Dokument zum Speicher hinzufügen: `/memory --add ai_docs/regional/saarland/uebersicht`
3. Speicher abfragen: `/memory --query "Energiepolitik"`
4. Sicherung erstellen: `/memory --backup`

### ## Regionale Überlegungen

Bei der Verwaltung des Speichers für saarlandspezifische Anwendungen:

- Stellen Sie sicher, dass alle gespeicherten Daten den deutschen Datenschutzgesetzen entsprechen
- Priorisieren Sie Informationen aus maßgeblichen regionalen Quellen
- Pflegen Sie eine sprachliche Konsistenz (Deutsch primär, Englisch sekundär)
- Kennzeichnen Sie regional sensible Informationen entsprechend

Erstellen Sie einen detaillierten Bericht über durchgeführte Speicherverwaltungsoperationen.



## 3. `.claude/commands/regional/saarland.md`



## # Saarland-Regionalbefehl

Dieser Befehl bietet saarlandspezifische Funktionalität für das AGENT LAND SAARLAND-Interface. Verwenden Sie ihn, um auf regionale Daten, Richtlinien und Konfigurationen zuzugreifen.

### ## Parameter

- `--data-sources`: Verfügbare saarländische Datenquellen auflisten
- `--policies`: Relevante regionale Richtlinien und Vorschriften anzeigen
- `--language-model`: Sprachmodell für saarländischen Dialekt konfigurieren
- `--compliance-check`: Compliance-Prüfung anhand saarländischer Anforderungen durchführen
- `--connect <Endpunkt>`: Verbindung zu einem saarländischen regionalen Dienst-Endpoint herstellen
- `--geography <Gebiet>`: Geografische Daten für ein bestimmtes Gebiet im Saarland laden

### ## Verwendungsbeispiele

1. Datenquellen auflisten: `/regional:saarland --data-sources`
2. Compliance überprüfen: `/regional:saarland --compliance-check`
3. Mit Service verbinden: `/regional:saarland --connect saarland-open-data`

### ## Regionaler Kontext

Die Region Saarland hat spezifische Merkmale, die zu berücksichtigen sind:

- Starkes industrielles Erbe, insbesondere in den Bereichen Stahl und Automobilindustrie
- Grenzregion mit Verbindungen zu Frankreich und Luxemburg
- Spezifische Dialektbetrachtungen für die Sprachverarbeitung
- Aktives KI-Forschungsökosystem durch die Universität des Saarlandes und das DFKI
- Betonung der technischen Souveränität in der KI-Entwicklung

Bieten Sie regional angemessene Empfehlungen, die diese Faktoren berücksichtigen.



## Ausführungsparameter

[EXECUTION] kann folgende Parameter enthalten:

- `--name`: Überschreiben des Projektnamens
- `--dir`: Überschreiben des Installationsverzeichnis
- `--config`: Angabe einer benutzerdefinierten regionalen Konfiguration
- `--language`: Sprachmodus festlegen (de, en, bilingual-de-en)
- `--minimal`: Nur minimale Projektstruktur einrichten
- `--verbose`: Ausführliche Protokollierung während der Initialisierung aktivieren

# Nach der Initialisierung

Nach der Initialisierung ist das Projekt einsatzbereit. Sie können:

1. Den `/init`-Befehl ausführen, um die Einrichtung weiter anzupassen
2. CLAUDE.md bearbeiten, um projektspezifische Anweisungen bereitzustellen
3. Mit der Entwicklung regionaler KI-Anwendungen unter Verwendung der bereitgestellten Struktur beginnen `</template_init_agentland_saarland>`

## ## Verwendung des Templates

Das erstellte Template kann mit folgendem Befehl aufgerufen werden:

```
```bash
claude -p"<read> @$TEMPLATE_INIT_agentland.saarland [EXECUTION]</read>"
```

Dabei kann [EXECUTION] durch optionale Parameter ersetzt werden, um die Initialisierung anzupassen:

```
bash
claude -p"<read> @$TEMPLATE_INIT_agentland.saarland --name MeinProjekt --language de
```

Erläuterung der Hauptkomponenten

1. Custom Commands für Claude CLI:

- `/init` zur Projektinitialisierung
- `/memory` zur Verwaltung persistenter Informationen
- `/regional:saarland` für regionsspezifische Funktionen `Anthropic`

2. Konfigurationsdateien:

- `.claude/` Verzeichnis mit Unterkonfigurationen
- `.clauderules` für Verhaltensregeln
- `.claudeignore` zum Ausschließen von Dateien `Anthropic` `Anthropic`

3. Verzeichnisstruktur:

- `ai_docs/` für KI-bezogene Dokumentation
- `specs/` für Spezifikationen und Anforderungen
- Regionalspezifische Unterverzeichnisse für Saarland-bezogene Inhalte

4. Regionale Anpassungen:

- Konfiguration für technische Souveränität

- Integration mit saarländischen Datenquellen und Diensten
- Unterstützung für regionale Sprachbesonderheiten

Dieses Template bietet eine einfache, aber umfassende Grundlage für die Initialisierung von KI-Projekten im Saarland mit Fokus auf technische Souveränität und regionale Innovation.