

WordPress & "KI"

Wie WordPress 7.0 "KI" zur Plattformfunktion macht

Hi 🖐️

Simon Kraft

simonkraft.de/hi

WordPress seit 2008

Meetups seit 2012

Produkt Manager @ Patchstack



"KI" war bisher Plugin-Chaos

WordPress 7.0
"KI" wird *Infrastruktur*

Das AI-Team

Gegründet Mai 2025

- James LePage (Automattic)
- Felix Arntz (Vercel)

Prinzip: *Canonical first, Core when ready*

make.wordpress.org/core/tag/ai

Die vier Building Blocks

1. PHP AI Client SDK
2. Abilities API
3. MCP-Adapter
4. AI Experiments Plugin

Baustein 1: PHP AI Client SDK

PHP AI Client SDK

Provider-agnostische Textgenerierung

```
$text = wp_ai_client_prompt(  
    'Fasse die Vorteile von Caching zusammen.'  
)  
->using_temperature( 0.7 )  
->generate_text();
```

Unterstützt: Anthropic · Google · OpenAI · und weitere
make.wordpress.org/core/tag/ai-client

Immer zuerst prüfen

```
$builder = wp_ai_client_prompt( 'test' );  
  
if ( $builder->is_supported_for_text_generation() ) {  
    // UI anzeigen  
}
```

**Konnektoren: Provider
verwalten**

Drei offizielle Provider in WP 7.0:

- AI Provider for Anthropic
- AI Provider for Google
- AI Provider for OpenAI

Baustein 2: Abilities API

Abilities API

WordPress-Funktionen maschinenlesbar machen

Seit WordPress 6.9

make.wordpress.org/core/tag/abilities-api

Ability registrieren

```
add_action( 'wp_abilities_api_init', function () {
    wp_register_ability(
        'mein-plugin/neueste-beitraege',
        [
            'label'      => 'Neueste Beiträge',
            'category'   => 'content-retrieval',
            'input_schema' => [
                'type'      => 'object',
                'properties' => [
                    'anzahl' => [
                        'type'      => 'integer',
                        'default' => 5,
                    ],
                ],
            ],
            'execute_callback'      => 'mab_get_neueste_beitraege',
            'permission_callback' => fn() => current_user_can( 'read' ),
            'meta' => [ 'show_in_rest' => true ],
        ]
    );
} );
```

Abilities + AI Client kombinieren

Hier wird's spannend

```
'execute_callback' => function ( $input ) {  
    $inhalt = wp_strip_all_tags(  
        get_post( $input['post_id'] )->post_content  
    );  
  
    return [  
        'zusammenfassung' => wp_ai_client_prompt(  
            "Fasse diesen Text in 2-3 Sätzen zusammen: $inhalt"  
        )  
        ->using_temperature( 0.3 )  
        ->using_max_tokens( 300 )  
        ->generate_text(),  
    ];  
}
```

Die Ability definiert die Schnittstelle.

Der AI Client übernimmt die Kommunikation.

Das Plugin kennt weder API-Key noch Anbieter.

Baustein 3: Model Context Protocol

MCP - ein offener Standard

"KI"-Assistent fragt WordPress:

„Was kannst du?“ → WordPress antwortet mit Abilities

modelcontextprotocol.io

Der WordPress MCP-Adapter

```
wordpress/mcp-adapter
```

MCP-Server ab Werk:

```
/wp-json/mcp/mcp-adapter-default-server
```

Drei Meta-Tools:

1. Abilities auflisten
2. Ability-Details abrufen
3. Ability ausführen

Noch kein Core-Bestandteil – separates Plugin
github.com/WordPress/mcp-adapter

Baustein 4: AI Experiments Plugin

Das kanonische Plugin des AI-Teams

Opt-in: jede Funktion einzeln aktivierbar

Voraussetzung: WordPress 7.0+

de.wordpress.org/plugins/ai

Einrichtung in 5 Schritten

1. API-Key holen → aistudio.google.com (kostenlos)
2. Einstellungen → Connectors → Google → Installieren
3. API-Key eintragen & speichern
4. Plugins → Installieren → „AI“ → Aktivieren
5. Einstellungen → AI → Titelgenerierung: Ein
Dann: Beitrag öffnen → Inhalt schreiben → Generate ✓

Jetzt selber bauen

```
add_action( 'wp_abilities_api_init', function () {
    wp_register_ability(
        'ki-beschreiber/plugin-beschreiben',
        [
            'execute_callback' => function ( $input ) {
                return [
                    'beschreibung' => wp_ai_client_prompt(
                        sprintf(
                            'Erkläre "%s" in einem Satz für Einsteiger. '
                            . 'Beschreibung: %s',
                            $input['plugin_name'],
                            $input['plugin_description']
                        )
                    )->using_max_tokens( 150 )->generate_text(),
                ];
            },
            'permission_callback' => fn() => current_user_can( 'activate_plugins' ),
            'meta' => [ 'show_in_rest' => true ],
        ]
    );
} );
```

Nutzbar via REST API, Abilities Explorer und Claude –
ohne weiteren Code.

Danke 

Fragen?