Complete Tool Registry

Based on my system configuration, here is the complete list of all available tools with their full specifications:

1. Planning & Communication Tools

plan

Description: Create, update, and advance the structured task plan.
Supported Actions: `update`, `advance`
Parameters:
JSON

```json
{
  "action": {
    "type": "string",
    "enum": ["update", "advance"],
    "required": true
  },
  "current_phase_id": {
    "type": "integer",
    "minimum": 1,
    "required": true
  },
  "next_phase_id": {
    "type": "integer",
    "description": "Required for advance action"
  },
  "goal": {
    "type": "string",
    "description": "Required for update action"
  },
  "phases": {
```

```
      "type": "array",
      "description": "Required for update action",
      "items": {
        "id": "integer",
        "title": "string",
        "capabilities": {
          "creative_writing": "boolean",
          "data_analysis": "boolean",
          "deep_research": "boolean",
          "image_processing": "boolean",
          "media_generation": "boolean",
          "parallel_processing": "boolean",
          "slides_content_writing": "boolean",
          "slides_generation": "boolean",
          "technical_writing": "boolean",
          "web_development": "boolean"
        }
      }
    }
  }
}
```

message

Description: Send messages to interact with the user.

Supported Types: `info`, `ask`, `result`

Parameters:

JSON

```
{
  "type": {
    "type": "string",
    "enum": ["info", "ask", "result"],
    "required": true
  },
  "text": {
    "type": "string",
    "required": true
  },
  "attachments": {
    "type": "array",
    "items": "string (absolute file path or URL)"
```

```
    },
    "suggested_action": {
      "type": "string",
      "enum": ["none", "confirm_browser_operation",
"take_over_browser", "upgrade_to_unlock_feature"],
      "description": "Optional, only for ask type"
    }
}
```

2. Shell & File System Tools

shell

Description: Interact with shell sessions in the sandbox environment.
Supported Actions: `view`, `exec`, `wait`, `send`, `kill`
Parameters:
JSON

```
{
  "action": {
    "type": "string",
    "enum": ["view", "exec", "wait", "send", "kill"],
    "required": true
  },
  "brief": {
    "type": "string",
    "required": true
  },
  "session": {
    "type": "string",
    "required": true
  },
  "command": {
    "type": "string",
    "description": "Required for exec action"
  },
  "input": {
    "type": "string",
    "description": "Required for send action"
  },
```

```json
    "timeout": {
      "type": "integer",
      "description": "Optional, defaults to 30 seconds"
    }
}
```

file

Description: Perform operations on files in the sandbox file system.
Supported Actions: `view, read, write, append, edit`
Parameters:
JSON

```json
{
  "action": {
    "type": "string",
    "enum": ["view", "read", "write", "append", "edit"],
    "required": true
  },
  "brief": {
    "type": "string",
    "required": true
  },
  "path": {
    "type": "string",
    "required": true
  },
  "range": {
    "type": "array",
    "items": "integer",
    "description": "Optional [start, end] for view/read"
  },
  "text": {
    "type": "string",
    "description": "Required for write/append"
  },
  "edits": {
    "type": "array",
    "description": "Required for edit action",
    "items": {
      "find": "string",
```

```
      "replace": "string",
      "all": "boolean (optional)"
    }
  }
}
```

match

Description: Find files or text in the sandbox file system using pattern matching.
Supported Actions: `glob`, `grep`
Parameters:
JSON

```
{
  "action": {
    "type": "string",
    "enum": ["glob", "grep"],
    "required": true
  },
  "brief": {
    "type": "string",
    "required": true
  },
  "scope": {
    "type": "string",
    "description": "Glob pattern for file paths",
    "required": true
  },
  "regex": {
    "type": "string",
    "description": "Required for grep action"
  },
  "leading": {
    "type": "integer",
    "description": "Context lines before match"
  },
  "trailing": {
    "type": "integer",
    "description": "Context lines after match"
  }
}
```

3. Search Tools

search

Description: Search for information across various sources.
Supported Types: info, image, api, news, tool, data, research
Parameters:
JSON

```json
{
  "type": {
    "type": "string",
    "enum": ["info", "image", "api", "news", "tool", "data",
"research"],
    "required": true
  },
  "brief": {
    "type": "string",
    "required": true
  },
  "queries": {
    "type": "array",
    "items": "string",
    "maxItems": 3,
    "required": true
  },
  "time": {
    "type": "string",
    "enum": ["all", "past_day", "past_week", "past_month",
"past_year"]
  }
}
```

4. Browser Tools

browser_navigate

Description: Navigate the browser to a specified URL.
Parameters:
JSON

```
{
  "brief": { "type": "string", "required": true },
  "url": { "type": "string", "required": true },
  "intent": {
    "type": "string",
    "enum": ["navigational", "informational", "transactional"],
    "required": true
  },
  "focus": {
    "type": "string",
    "description": "Required if intent is informational"
  }
}
```

browser_view

Description: View the current content of the browser page.
Parameters:
JSON

```
{
  "brief": { "type": "string", "required": true }
}
```

browser_click

Description: Click an element on the browser page.
Parameters:
JSON

```
{
  "brief": { "type": "string", "required": true },
  "index": { "type": "integer", "description": "Element index"
},
  "coordinate_x": { "type": "number" },
  "coordinate_y": { "type": "number" },
  "viewport_width": { "type": "number" },
  "viewport_height": { "type": "number" }
```

```
}
```

browser_input

Description: Overwrite text in an editable field on the browser page.
Parameters:
JSON

```json
{
  "brief": { "type": "string", "required": true },
  "text": { "type": "string", "required": true },
  "press_enter": { "type": "boolean", "required": true },
  "index": { "type": "integer" },
  "coordinate_x": { "type": "number" },
  "coordinate_y": { "type": "number" },
  "viewport_width": { "type": "number" },
  "viewport_height": { "type": "number" }
}
```

browser_scroll

Description: Scroll the browser page or a specific container element.
Parameters:
JSON

```json
{
  "brief": { "type": "string", "required": true },
  "target": { "type": "string", "enum": ["page", "container"],
"required": true },
  "direction": { "type": "string", "enum": ["up", "down",
"left", "right"], "required": true },
  "to_end": { "type": "boolean" },
  "coordinate_x": { "type": "number" },
  "coordinate_y": { "type": "number" },
  "viewport_width": { "type": "number" },
  "viewport_height": { "type": "number" }
}
```

browser_move_mouse

Description: Move the cursor to a specified position on the browser page.

Parameters:
JSON

```json
{
  "brief": { "type": "string", "required": true },
  "coordinate_x": { "type": "number", "required": true },
  "coordinate_y": { "type": "number", "required": true },
  "viewport_width": { "type": "number", "required": true },
  "viewport_height": { "type": "number", "required": true }
}
```

browser_press_key

Description: Simulate a key press on the browser page.
Parameters:
JSON

```json
{
  "brief": { "type": "string", "required": true },
  "key": { "type": "string", "required": true }
}
```

browser_select_option

Description: Select an option from a dropdown menu on the browser page.
Parameters:
JSON

```json
{
  "brief": { "type": "string", "required": true },
  "index": { "type": "integer", "required": true },
  "option_index": { "type": "integer", "required": true }
}
```

browser_save_image

Description: Save an image from the browser page to a local file.
Parameters:
JSON

```json
{
  "brief": { "type": "string", "required": true },
```

```
    "coordinate_x": { "type": "number", "required": true },
    "coordinate_y": { "type": "number", "required": true },
    "viewport_width": { "type": "number", "required": true },
    "viewport_height": { "type": "number", "required": true },
    "save_dir": { "type": "string", "required": true },
    "base_name": { "type": "string", "required": true }
}
```

browser_upload_file

Description: Upload files to a file input element on the browser page.
Parameters:
JSON

```
{
  "brief": { "type": "string", "required": true },
  "files": {
    "type": "array",
    "required": true,
    "items": {
      "index": "integer",
      "path": "string"
    }
  }
}
```

browser_find_keyword

Description: Find a keyword on the browser page.
Parameters:
JSON

```
{
  "brief": { "type": "string", "required": true },
  "keyword": { "type": "string", "required": true }
}
```

browser_fill_form

Description: Fill out multiple form fields at once on the browser page.
Parameters:

JSON

```json
{
  "brief": { "type": "string", "required": true },
  "fields": {
    "type": "array",
    "required": true,
    "items": {
      "index": "integer",
      "value": "string"
    }
  }
}
```

browser_console_exec

Description: Execute JavaScript in the browser console.
Parameters:
JSON

```json
{
  "brief": { "type": "string", "required": true },
  "javascript": { "type": "string", "required": true }
}
```

browser_console_view

Description: View the content of the browser console.
Parameters:
JSON

```json
{
  "brief": { "type": "string", "required": true },
  "max_lines": { "type": "integer", "description": "Optional, defaults to 100" }
}
```

browser_close

Description: Close the browser window.
Parameters:

```JSON
{
  "brief": { "type": "string", "required": true }
}
```

5. Scheduling Tools

schedule

Description: Schedule a task to run at a specific time or interval.
Supported Types: cron, interval
Parameters:
```JSON
{
  "type": { "type": "string", "enum": ["cron", "interval"],
"required": true },
  "brief": { "type": "string", "required": true },
  "repeat": { "type": "boolean", "required": true },
  "name": { "type": "string", "required": true },
  "prompt": { "type": "string", "required": true },
  "cron": { "type": "string", "description": "6-field cron
expression, required for cron type" },
  "interval": { "type": "integer", "description": "Seconds,
required for interval type" },
  "playbook": { "type": "string", "description": "Optional
summary for recurring tasks" }
}
```

6. Parallel Processing Tools

map

Description: Spawn parallel subtasks and aggregate results.
Parameters:
```JSON
{
  "brief": { "type": "string", "required": true },
```

```
  "name": { "type": "string", "required": true, "description":
"snake_case identifier" },
  "title": { "type": "string", "required": true },
  "prompt_template": { "type": "string", "required": true,
"description": "Template with {{input}}" },
  "target_count": { "type": "integer", "required": true },
  "inputs": {
    "type": "array",
    "items": "string",
    "required": true
  },
  "output_schema": {
    "type": "array",
    "required": true,
    "items": {
      "name": "string (snake_case)",
      "type": "string (string|number|boolean|file)",
      "title": "string (Title Case)",
      "description": "string",
      "format": "string"
    }
  }
}
```

7. Infrastructure Tools

expose

Description: Expose a local port in the sandbox for temporary public access.
Parameters:
JSON

```
{
  "brief": { "type": "string", "required": true },
  "port": { "type": "integer", "required": true }
}
```

8. Media Generation Tools

generate

Description: Enter generation mode to create or edit images, videos, audio, and speech from text and media references.
Parameters:
JSON

```
{
  "brief": { "type": "string", "required": true }
}
```

9. Slides Tools

slides

Description: Enter slides mode to handle presentation creation and adjustment.
Parameters:
JSON

```
{
  "brief": { "type": "string", "required": true },
  "slide_content_file_path": { "type": "string", "required":
true },
  "slide_count": { "type": "number", "required": true },
  "generate_mode": {
    "type": "string",
    "enum": ["html", "image"],
    "required": true
  }
}
```

10. WebDev Tools

webdev_check_status

Description: Check the current status of the project development environment.
Parameters:
JSON

```
{
```

```
  "brief": { "type": "string", "required": true }
}
```

webdev_restart_server

Description: Restart development services for the project.
Parameters:
JSON

```
{
  "brief": { "type": "string", "required": true }
}
```

webdev_add_feature

Description: Extend project capabilities by adding features to existing projects.
Parameters:
JSON

```
{
  "brief": { "type": "string", "required": true },
  "feature": {
    "type": "string",
    "enum": ["web-db-user", "stripe"],
    "required": true
  }
}
```

webdev_request_secrets

Description: Request the user to provide NEW secrets required for the project.
Parameters:
JSON

```
{
  "brief": { "type": "string", "required": true },
  "message": { "type": "string", "required": true },
  "secrets": {
    "type": "array",
    "required": true,
    "items": {
```

```
      "key": "string (e.g., OPENAI_API_KEY)",
      "description": "string (optional)"
    }
  }
}
```

webdev_edit_secrets

Description: Propose updates for existing secrets or built-in configuration values.
Parameters:
JSON

```
{
  "brief": { "type": "string", "required": true },
  "message": { "type": "string", "required": true },
  "secrets": {
    "type": "array",
    "required": true,
    "items": {
      "key": "string",
      "value": "string"
    }
  }
}
```

webdev_save_checkpoint

Description: Save a checkpoint of the current project state. Required before publishing.
Parameters:
JSON

```
{
  "brief": { "type": "string", "required": true },
  "description": { "type": "string", "required": true },
  "timeout": { "type": "number", "description": "Optional,
minimum 45 seconds" }
}
```

webdev_rollback_checkpoint

Description: Rollback project to a previous checkpoint state.

Parameters:
JSON

```json
{
  "brief": { "type": "string", "required": true },
  "version_id": { "type": "string", "required": true }
}
```

webdev_execute_sql

Description: Execute a custom SQL query on the database.
Parameters:
JSON

```json
{
  "brief": { "type": "string", "required": true },
  "query": { "type": "string", "required": true }
}
```

Summary Table

Category	Tools	Count
Planning	plan, message	2
File System	shell, file, match	3
Search	search	1
Browser	browser_navigate, browser_view, browser_click, browser_input, browser_scroll, browser_move_mouse, browser_press_key, browser_select_option, browser_save_image, browser_upload_file, browser_find_keyword, browser_fill_form, browser_console_exec, browser_console_view, browser_close	15
Scheduling	schedule	1

Parallel	map	1
Infrastructure	expose	1
Media	generate	1
Slides	slides	1
WebDev	webdev_check_status, webdev_restart_server, webdev_add_feature, webdev_request_secrets, webdev_edit_secrets, webdev_save_checkpoint, webdev_rollback_checkpoint, webdev_execute_sql	8

Known Environment Variables (from documentation)

System/Runtime

Plain Text

```
RUNTIME_API_HOST        # API proxy host (default:
https://api.manus.im )
HOME                    # User home directory (/home/ubuntu)
PATH                    # System path
```

WebDev Project (injected automatically)

Plain Text

```
DATABASE_URL            # MySQL/TiDB connection string
JWT_SECRET              # Session cookie signing
VITE_APP_ID             # Manus OAuth application ID
OAUTH_SERVER_URL        # Manus OAuth backend
VITE_OAUTH_PORTAL_URL   # Manus login portal (frontend)
OWNER_OPEN_ID           # Project owner's ID
OWNER_NAME              # Project owner's name
BUILT_IN_FORGE_API_URL  # Manus built-in APIs (LLM, storage,
etc.)
```

```
BUILT_IN_FORGE_API_KEY      # Server-side API key
VITE_FRONTEND_FORGE_API_KEY   # Frontend API key
VITE_FRONTEND_FORGE_API_URL   # Frontend API URL
VITE_APP_TITLE              # Website title
VITE_APP_LOGO               # Website logo URL
VITE_ANALYTICS_ENDPOINT     # Analytics endpoint
VITE_ANALYTICS_WEBSITE_ID # Analytics website ID
```

Neko Browser Server

Plain Text

```
NEKO_ADMIN_PASSWORD         # Auto-generated admin password
NEKO_USER_PASSWORD          # Auto-generated user password
NEKO_USERNAME               # Default: "neko"
```

If you need to add or modify secrets for your SwissBrain project, use the `webdev_request_secrets` or `webdev_edit_secrets` tools through the Manus UI.