

附件 3

国土空间基础信息平台众筹工具 技术要求 (试 行)

二〇二四年五月

目 次

引 言.....	VI
1 范围	1
2 工具定义	1
3 工具类型	1
3.1 API 工具	1
3.2 WEB 页面工具	1
3.3 SDK 工具	1
4 总体要求	1
4.1 运行环境要求	1
4.1.1 信创软硬件	1
4.1.2 GIS 软件.....	2
4.1.3 数据库管理系统	2
4.2 数据要求	2
4.2.1 数据清单	2
4.2.2 数据输入	3
4.2.3 数据访问	4
4.2.4 数据传输	5
4.2.5 数据输出	5
4.3 部署与运维要求	5
4.3.1 版本管理	5
4.3.2 版本更新要求	5
4.3.3 部署模式	6
4.3.4 配置要求	7
4.3.5 运行与监控	7
5 API 工具开发要求.....	7
5.1 技术要求	7
5.1.1 访问协议	7
5.1.2 数据交换格式	7
5.1.3 字符编码	7
5.1.4 参数规范	7
5.1.5 接口调用规范	7
5.1.6 接口返回规范	7
5.2 接口要求	8
5.2.1 接口路径规则	8
5.2.2 接口清单	8

5.2.3	标准接口	8
5.2.4	自定义接口	13
5.2.5	任务接口	13
5.3	工具测试	15
5.4	工具编译与打包要求	16
6	WEB 页面工具开发要求	16
6.1	技术要求	16
6.1.1	浏览器兼容要求	16
6.1.2	字符编码	16
6.1.3	协议类型	16
6.1.4	URL 命名规范	17
6.1.5	HTML 规范	17
6.2	前端页面开发要求	17
6.2.1	界面设计规范	17
6.2.2	前端开发规范	18
6.3	后端服务开发要求	19
6.4	工具测试	19
6.5	工具编译与打包要求	20
7	SDK 工具开发要求	20
7.1	总体要求	20
7.2	技术要求	20
7.2.1	类库设计规范	20
7.2.2	参数规范	20
7.2.3	调用规范	21
7.2.4	错误处理和调试	21
7.3	SDK 内容要求	21
7.3.1	类库	21
7.3.2	方法	21
7.4	工具测试	21
7.5	编译与打包要求	22
8	工具安全要求	22
8.1	技术框架安全	22
8.2	代码安全	22
8.3	数据安全	22
8.4	使用安全	22
8.5	日志与审计	23
9	工具发布与共享要求	24

9.1	工具成果包	24
9.2	工具发布流程	24
9.2.1	工具提供方	25
9.2.2	工具管理方	25
9.3	工具共享流程	26
9.3.1	工具使用方	26
9.3.2	工具提供方	26
9.3.3	工具管理方	27
9.4	附表 国土空间基础信息平台工具注册表.....	28

引 言

为贯彻落实《自然资源数字化治理能力提升总体方案》（自然资发〔2024〕33号）中“提高国土空间基础信息平台智能化水平”要求，开展工具众筹建设，面向自然资源部系统单位、地方自然资源主管部门，建设一批通用性强、易用性好、复用率高的工具，建立即插即用、按需扩展、便捷迁移的技术机制，为底线守护、格局优化、绿色低碳、权益维护场景构建提供高效、智能的共性支撑服务能力，推进构建“共建、共享、共用”的众筹模式，形成央地共建、一地创新、全国共享的数字生态。

1 范围

本文档描述了国土空间基础信息平台（以下简称“平台”）工具接入要求、发布与共享要求和安全要求等，为工具提供方依托平台工具中心注册、接入、上架和发布工具提供指导。

2 工具定义

工具是实现特定功能的软件，支撑具体场景构建、具备输入输出功能、具有较高通用性和可移植性的 API 工具、Web 页面工具和 SDK 工具。

3 工具类型

工具具有较强的通用性、独立性，基于平台支撑各类业务场景快速构建，具有较高的可复用性。根据工具的技术形态包括三类工具：API 工具、Web 页面工具和 SDK 工具。

3.1 API 工具

API 工具是一种或一组通过 API（应用程序接口）来完成相关任务的软件或服务。API 工具的使用对象是开发人员，帮助开发人员在创建、测试、管理应用过程中提高软件复用性、提升工作效率和质量。

3.2 Web 页面工具

Web 页面工具是包含用户交互界面，可以实现特定业务功能的软件或服务。Web 页面工具的使用对象是业务人员，依托场景开展应用。Web 页面工具无需二次开发，相对独立，通过部署和配置可直接复用。

3.3 SDK 工具

SDK 工具是一种或一组软件开发工具（软件开发工具包）。SDK 工具的使用对象是开发人员，开发人员基于各自开发框架引入该工具包进行二次开发，实现对工具通用能力的复用。

4 总体要求

4.1 运行环境要求

4.1.1 信创软硬件

工具推荐在信创软硬件环境下部署和运行。需明确说明已适配的处理器、操作系统、数据库管理系统、GIS 及其他中间件、浏览器等内容产品名称和版本号。

4.1.2 GIS 软件

涉及 GIS 计算的工具，在设计实现时优先考虑与特定 GIS 产品进行解耦，保障工具的通用性和易复用性。若涉及专业的 GIS 计算，必须依赖特定 GIS 产品，在工具注册时需要明确所使用的 GIS 产品名称及版本号。

4.1.3 数据库管理系统

涉及依赖数据库的工具，在数据输入和输出环节，原则上建议只使用数据库管理系统的基础功能，如数据定义、数据操作和数据库访问接口等。若该工具依赖特定的数据库管理系统，在工具注册时需要明确所使用的数据库管理系统的名称及版本号。

4.2 数据要求

工具提供方应明确工具所需数据内容和标准结构，同时应考虑数据在使用过程中的安全性，需要从数据输入到数据输出每个环节约束数据的连接、存储和使用方式。

4.2.1 数据清单

根据工具所需的数据内容编写数据清单，详细描述需要的数据内容以及每项数据的标准结构。

4.2.1.1 数据内容

工具提供方应明确数据清单中每项数据的名称、用途、支持的输入格式，以及该数据是否为必须项等内容。同时，针对工具使用方对计算结果进行本地化存储的需求，将计算结果数据列入到数据清单中。

表 4-1 数据清单

序号	数据项名称	数据项说明	数据用途	数据格式	约束条件	数据流向类型
注 1：数据项名称是指所需数据的名称，依据自然资源“一张图”数据资源目录的名称； 注 2：数据项说明是对该数据项的业务描述，便于用户准确理解所需数据； 注 3：数据用途是说明该数据项在本工具中发挥的作用； 注 4：数据格式是指该数据项输入或输出的保存形态，如数据库表、文件格式等，如果支持多种需要都列出来； 注 5：约束条件取值：M（必选）、O（可选）、C（条件必选）。 注 6：数据流向类型包括输入和输出两类。						

4.2.1.2 数据结构

根据工具所需数据清单，针对每一项数据，应详细给出标准的数据结构，包括每个字段的名称、编码、字段类型、字段长度、小数位数、值域范围和是否必填等约束

条件。

表 4-2 数据结构

序号	字段名称	字段代码	字段类型	字段长度	小数位数	值域	约束条件	备注

4.2.2 数据输入

工具支持三种数据输入方式，第一种以字符串/流方式输入数据内容，空间数据需采用 GeoJson、WKT 格式或 GML 格式；第二种传递数据文件的访问地址，空间数据需采用 shp、tif 等格式；第三种提供数据库连接信息，包括数据库地址和用户信息。对于第三种方式，需考虑数据库的安全性，不能直接将数据库连接信息以明文方式进行传输，需要对数据库的连接信息进行安全设计，保障源数据库的安全性。同时，为了提升工具的易用性，比如 Web 页面工具，应将必要的数据输入在工具配置文件体现，不可直接写入工具操作界面。

4.2.2.1 操作步骤

1. 在工具的配置文件中保存获取数据库连接信息的 API 接口地址；
2. 在工具内调用该 API 接口获取数据库连接信息；
3. 连接到数据库进行数据操作。

4.2.2.2 标准接口设计

获取数据库连接信息的标准 API 由平台统一提供，接口设计如下：

1. 使用场景

获取工具所需要的数据库连接信息，保证每个工具在平台授权下连接各数据库。

2. 接口详情

接口名称：getDBConnection；

接口路径：/platform/coreApi/getDBConnection；

调用方式：GET 接口；

返回格式：JSON。

3. 输入参数

输入参数主要是访问 API 接口的 token 验证码，工具完成本地部署后，由平台提供，写入到工具的配置文件中。

表 4-3 输入参数表

序号	参数名称	参数类型	参数说明	是否必填	备注
1	token	String	授权码	是	

4. 返回结果

API 接口返回结果包括状态码、消息内容、是否调用成功的状态以及具体的数据库连接信息，详细结构及字段说明如下表所示：

表 4-4 返回结果结构表

序号	参数名称	子参数名称	参数类型	参数说明
1	code	-	String	状态码，如 200 表示正常，400 表示异常
2	message	-	String	返回消息
3	success	-	Boolean	接口调用成功标识
4	data	-	Array	返回结果对象、数组类型，支持返回多个数据库连接信息
5		ip	String	数据库服务器 IP
6		port	String	数据库端口号
7		schema	String	数据库模式名
8		username	String	用户名
9		password	String	密码
10		driverClassName	String	数据库驱动类型名称

5. 返回示例

```

{
  "code":200,
  "message":"请求成功",
  "success":true,
  "data":{
    "ip":"127.0.0.1",
    "port":"5432",
    "schema":"public",
    "username":"postgres",
    "password":"postgres",
    "driverClassName":"org.postgresql.Driver"
  }
}

```

4.2.3 数据访问

工具通过只读方式访问平台授权的数据源，读取数据后进行计算和处理，不能对

原始数据进行编辑操作。在工具开发时应该严格检查代码逻辑，确保原始数据安全。

4.2.4 数据传输

为保证数据安全性，若涉及敏感数据，在工具输出数据传输过程中需要采用国密算法进行加密处理。工具提供方选择合适的加密算法，如 SM2 或 SM4 等，对输出数据进行加密处理，并将密钥写到工具部署包元数据描述文件中；工具使用方从部署包中获取密钥信息，并在应用层面进行解密使用。

4.2.5 数据输出

工具的过程数据不存储在数据库，而是使用临时文件方式在工具内部流转，使用后清除。

工具应至少提供以下一种方式支持存储计算结果。第一种方式是直接将计算结果作为返回值传给调用方，空间数据需采用 GeoJson 格式；第二种方式是将计算结果的存储位置作为一个参数传递至工具，工具按照参数值（数据库连接、文件存储路径）将计算结果写入到目标位置；第三种方式是在工具使用过程中，提供结果导出功能，便于用户直接操作。

4.3 部署与运维要求

4.3.1 版本管理

工具应按版本进行管理，工具发布时应明确版本号，允许后续进行版本更新。

工具的版本号一般由多个部分组成，格式为 X.Y.Z，其中 X、Y 和 Z 分别代表主版本号、次版本号和修订号。

主版本号：当工具的大量功能发生重大改变或者升级时，主版本号会发生改变。例如，从 V1 升级到 V2。

次版本号：当工具增加新的功能或者改进现有功能时，次版本号会发生改变。例如，从 V1.0 升级到 V1.1。

修订号：当工具修复程序中的错误或者问题时，修订号会发生改变。例如，从 V1.1.0 升级到 V1.1.1。

4.3.2 版本更新要求

工具提供方对已注册到平台的工具进行版本更新，在版本更新过程中需要遵循以下要求：

1. 版本更新流程按照工具发布流程重新进行发布，详细情况参照 9.2，工具管理方对工具成果包重新进行安全和稳定性检查和测试。

2. 新版本应向下兼容，不能随意修改以往版本中暴露方法的类型、个数和结构，包括方法命名、参数及返回结构等。当需要对上述内容进行修改时，应设计新的接口或方法。

3. 对有多个版本的工具，工具提供方应在最新版本中涵盖版本升级日志记录，方便工具使用方了解工具升级内容。

4.3.3 部署模式

4.3.3.1 部署方式

需要部署的工具应至少支持以下一种部署模式，包括集中部署、提供方部署和使用方部署。根据工具的应用场景和对数据的依赖程度选择相应的部署方式。

1. 集中部署：在平台中集中部署工具，提供在线服务。此类工具一般是依赖全国范围内的数据，通过在部平台集中部署，以在线服务方式供各地方用户使用，可最大化提升工具复用简易程度和效率。

2. 提供方部署：工具提供方在本地环境中部署工具，将工具的服务信息注册到平台工具中心，提供在线服务。

3. 使用方部署：工具使用方在本地环境中部署工具，提供离线服务。

4.3.3.2 部署方案

按照工具部署要求准备基础环境完成工具部署。

1. 若采用虚拟机部署，工具部署流程如下：

a) 申请虚拟机资源，并安装好基础环境，包括操作系统、CPU、磁盘、网络等，需与工具部署要求保持一致；

b) 将工具部署包拷贝到虚拟机中，并安装好工具所需依赖软件；

c) 根据工具部署手册，修改工具配置信息，完成本地化连接；

d) 启动工具并测试；

e) 使用工具。

2. 若采用容器部署，工具部署流程如下：

a) 申请容器运行宿主机，确保其已安装好容器基础环境，容器版本需与工具部署要求保持一致；

b) 将工具部署包拷贝到宿主机中，导入容器镜像并创建容器服务；

c) 根据工具部署手册，修改工具配置信息，完成本地化连接；

d) 启动工具并测试；

e) 使用工具。

4.3.4 配置要求

1. 应将工具配置与代码分离，使用配置文件或环境变量管理配置项，便于部署时调整；

2. 微服务架构的工具通过配置中心对配置进行统一管理，并尽量提供可视化的配置管理界面。

4.3.5 运行与监控

工具应记录运行过程中的操作及异常日志，并提供日志查询方法。

5 API 工具开发要求

5.1 技术要求

5.1.1 访问协议

采用 http 接口协议。

5.1.2 数据交换格式

接口请求及返回的数据应采用 JSON 或 XML 格式。

5.1.3 字符编码

接口请求及返回的数据应采用 UTF-8 编码。

5.1.4 参数规范

在设计接口参数时，应当明确说明每个参数的名称、类型、默认值、是否必填等内容，如参数为枚举值，需附上字典表。

5.1.5 接口调用规范

接口调用应采用 GET、POST 两种方式，禁止使用 PUT、DELETE 等方式。如参数值中包含特殊字符（如空格、+、#等），应该进行 URL 编码。

5.1.6 接口返回规范

工具提供方应限制接口单次调用返回数据的最大量，不超过 1000 条。

表 5-1 返回参数表

序号	参数名称	参数说明	参数类型
1	code	状态码（支持 200 表示请求成功、400 表示请求失败等）	String
2	message	服务器端请求响应消息	String
3	success	接口调用成功标识（true—成功、false—失败）	Boolean

4	data	响应业务结果数据，详见各服务接口	Object
---	------	------------------	--------

示例：

```

成功返回
{
  "code": "200",
  "message": "请求成功",
  "success": true,
  "data": {
    "name": "张三",
    "age": 30
  }
}

失败返回
{
  "code": 400,
  "message": "参数检验失败，请检查！",
  "success": false,
  "data": null
}

```

5.2 接口要求

API 工具提供一个或一组服务接口来实现业务逻辑，需提供服务接口清单。

5.2.1 接口路径规则

工具提供的服务接口访问路径应保持一致，接口名称需做区分，规则如下：“/工具名称/接口类别/请求操作”，如：/xxxTool/yyyApi/zzzOperate，其中工具名称统一以 Tool 作为后缀。

5.2.2 接口清单

API 工具注册到平台时应提交完整的接口清单，包括接口类别、接口名称、功能描述、访问路径等内容，其中接口类别按照工具所包含的功能进行分类，要求如下：

表 5-2 接口清单表

序号	接口类别	接口名称	功能描述	访问路径
1				
2				

5.2.3 标准接口

标准接口为每个 API 工具必须实现的接口，包括描述说明、接口列表、工具更新日志、健康状态检查等接口。

5.2.3.1 描述说明接口

主要包含工具的描述说明等基本信息，平台及其他调用方可调取该接口并同步工具的相关描述说明信息，实现与平台工具中心相关功能的自动衔接。

1. 接口详情

- a) 接口名称: getDescription;
- b) 接口路径: /xxxTool/coreApi/getDescription;
- c) 调用方式: GET;
- d) 返回格式: JSON。

2. 输入参数

无。

3. 返回结果

表 5-3 返回结果结构表

参数名	子参数名	类型	说明
code	-	Integer	状态码
message	-	String	返回消息
success	-	Boolean	接口调用成功标识
data	-	Object	返回结果对象
	name	String	工具名称
	description	String	工具的详细介绍描述
	effect	String	应用成效（可选）

返回示例:

```
{
  "code" :200,
  "message" : "请求成功",
  "success":true,
  "data" :
  {
    "name": "xxx 工具",
    "description" : "xxx 工具包含以下功能，可以用于 xx 用途，包含 xx 数据的展示和分析",
    "effec" : "xxx 工具在 xx 领域、xx 方面得到了推广应用，产生了 xx 成效。"
  }
}
```

5.2.3.2 接口列表

平台及其他调用方调取该接口以同步工具接口列表信息。如果该接口是通过页面

组件方式调用，应在返回结果参数 description 中进行特殊说明。

1. 接口详情

- a) 接口名称: getCapabilities;
- b) 接口路径: /xxxTool/coreApi/getCapabilities;
- c) 调用方式: GET;
- d) 返回格式: JSON。

2. 输入参数

无。

3. 返回结果

表 5-4 接口列表返回结果表

参数名	子参数名	类型	说明
code	-	Integer	状态码
message	-	String	返回消息
success	-	Boolean	接口调用成功标识
data	-	Object	返回结果
	apiList	Array	api 接口列表
	componentList	Array	页面组件接口列表
以下为接口对象内的属性			
	name	String	接口名称
	description	String	接口描述
	url	String	接口路径
	method	String	调用方法 get / post
	contentType	String	输入数据的格式 (get 类型时候可为空)
	input	String	输入参数数据结构规则
	output	String	返回参数数据结构规则

返回示例:

```

{
  "code" :200,
  "message" : "请求成功",
  "success":true,
  "data" :
  {
    "apiList" :[
      {
        "name" : "查询 xx 信息接口",
        "description" : "通过本接口可查询相关 xx 信息, 结果以 xx 形式返

```

```

    "url" :"/xxxTool/coreApi/getXxxinfo",
    "method" : "post",
    "contentType" : "application/json",
    "input" : "传入 json 对象，包括 name 属性和 type 属性",
    "output" : "返回查询结果数组，每个对象中包含以下属性..."
  }
],
"componentList": [
  {
    "name" : "遥感影像历史变化组件",
    "description": "通过嵌入本页面组件，传入位置图形可查看指定位置遥感影像变化",
    "url" : "/xxxTool/ygjc/ygyxComponent",
    "method" : "get",
    "input": "传入 areaId 用于标志图形 id；传入 geometryFormat 标识坐标字符串格式，可选择 wkt 或 geojson；geometry 传入空间图形。",
    "output" : "页面中渲染出传入图形，并将地图定位至该图形。"
  }
]
}
}

```

5.2.3.3 工具更新日志接口

主要包含接口更新相关的描述。平台及其他调用方可调取该接口用于同步工具的更新日志信息。

1. 接口详情

- a) 接口名称：getUpdateLog；
- b) 接口路径：/xxxTool/coreApi/getUpdateLog；
- c) 调用方式：GET；
- d) 返回格式：JSON。

2. 输入参数

无。

3. 返回结果

表 5-5 更新日志返回结果表

参数名	子参数名	类型	说明
code	-	Integer	状态码
message	-	String	返回消息
success	-	Boolean	接口调用成功标识
data	-	Array	返回结果数组

	version	String	版本名称
	description	String	版本更新内容描述
	updateTime	String	更新时间 yyyy-mm-dd (如 2023-02-27)

返回示例:

```
{
  "code":200,
  "message":"请求成功",
  "success":true,
  "data":[
    {
      "vernum":"V1.1",
      "description":"1. 新增了工具中的 xx 功能; 2. 完善了工具中的 xx 功能; 3. 修改了工具中的 xx 功能",
      "updateTime":"2023-02-27"
    }
  ]
}
```

5.2.3.4 健康状态检查接口

主要包含工具可用性检测、各接口可用性检查接口地址等信息，平台及其他调用方可调取该接口获取工具的健康检测信息，平台工具中心依据此接口对工具进行健康检查。

1. 接口详情

- a) 接口名称: getHealthStatus;
- b) 接口路径: /xxxTool/coreApi/getHealthStatus;
- c) 调用方式: GET;
- d) 返回格式: JSON。

2. 输入参数

无。

3. 返回结果

表 5-6 健康检测返回结果表

参数名	子参数名	类型	说明
code	-	Integer	状态码
message	-	String	返回消息
success	-	Boolean	接口调用成功标识
data	-	Array	返回结果数组
	name	String	健康检测项名称

	description	String	健康监测项的详细说明
	url	String	健康检测地址（返回 http 状态码 200 表示健康）

返回示例：

```

{
  "code" :200,
  "message" :“请求成功”,
  "success":true,
  "data" :[
    {
      "name" :“工具访问性验证”,
      "description" :“用于验证工具是否可以正常地访问，是否存在离线问题。”,
      "url" :“https://xxx.com/xxx”
    }
  ]
}

```

5.2.4 自定义接口

按照 5.1 章节的基本要求，根据工具功能需要定制开发其他接口。

5.2.5 任务接口

工具提供方可提供同步接口和异步接口。同步接口一般直接发起任务执行，参照 5.2.2 接口清单进行调用。异步接口名称一般以 Job 作为结尾，接口名称格式为：xxxJob。异步接口可分拆为提交任务接口和任务状态查询接口，其中涉及到的参数字典如下：

表 5-7 参数字典表

名称	参数名称	参数类型	参数说明
任务 ID	jobId	String	任务唯一 ID, 不可重复。
任务状态	jobStatus	String	任务状态，分为“成功”（success）、“失败”（failed）、“执行中”（executing）
任务进度	jobCompletion	float	任务完成进度百分比，取值范围 0~ 1，展示的时候转换为百分比。若无法计算准确百分比可以按执行流程返回粗略进度。

5.2.5.1 提交任务接口

提交任务接口接收输入参数，返回任务 ID。

1. 接口详情

a) 接口名称：submitJob；

- b) 接口路径: /xxxTool/yyyApi/zzzOperate/submitJob;
- c) 调用方式: POST;
- d) 返回格式: JSON。

2. 输入参数

具体参数以 JSON 对象传递给接口。

```
{
  "xxx1": "xxx1",
  "xxx2": "xxx2"
}
```

3. 返回结果

表 5-8 提交任务返回结果表

参数名	子参数名	类型	说明
code	-	Integer	状态码
message	-	String	返回消息
success	-	Boolean	接口调用成功标识
data	-	Object	返回结果对象
	jobId	String	任务 id

返回示例:

```
{
  "code" :200,
  "message" : "请求成功",
  "success": true,
  "data" : {
    "jobId" : "j164188d71c1c40d4bef4c931017d9051"
  }
}
```

5.2.5.2 任务状态查询接口

输入任务 ID 查询对应任务的执行状态信息，任务完成时返回状态码和执行结果。

1. 接口详情

- a) 接口名称: queryJob;
- b) 接口路径: /xxxTool/yyyApi/zzzOperate/queryJob;
- c) 调用方式: GET;
- d) 返回格式: JSON。

2. 输入参数

```
{  
  "jobId": "j164188d71c1c40d4bef4c931017d9051"  
}
```

3. 返回结果

表 5-9 任务状态查询返回结果表

参数名	子参数名	类型	说明
code	-	Integer	状态码
message	-	String	返回消息
success	-	Boolean	接口调用成功标识
data	-	Object	返回结果对象
	jobId	String	任务 id
	jobStatus	String	任务状态, 分为“success”(成功)、“failed”(失败)、“executing”(执行中)。
	jobMessage	Array	任务进度消息数组
	jobCompletion	Float	任务完成进度百分比, 取值范围 0~1, 展示时转换为百分比。若无法计算准确百分比可以按执行流程返回粗略进度。
	jobResult	Object	任务分析结果

返回示例:

```
{  
  "code" :200,  
  "message" : "请求成功",  
  "success":true,  
  "data" : {  
    "jobId" : "j164188d71c1c40d4bef4c931017d9051",  
    "jobStatus" : "failed",  
    "jobMessage" : [  
      "任务已提交",  
      "正在进行第一步模型计算",  
      "任务计算失败"  
    ],  
    "jobCompletion" :0.8,  
    "jobResult" : {}  
  }  
}
```

5.3 工具测试

1. 工具开发完成后, 参照 5.1 技术要求对工具成果进行检查和测试, 确保符合上

述技术要求。

2. 对工具进行功能测试，确保工具能够正常使用，计算结果准确性达到预期。

3. 对工具进行并发测试，详细记录工具部署环境的资源配置，以及不同并发访问数下的响应情况。

4. 对工具成果包进行安全扫描，在测试文档中记录扫描检查的内容项及检查结果，确保不存在已知安全漏洞。

5. 需要部署的工具在重新部署后，需根据工具使用手册说明，重新进行功能测试。

6. 整个工具测试完成后，需要将测试过程、测试结果及环境配置等信息记录到测试文档中，形成完整的测试报告。

5.4 工具编译与打包要求

1. 需要将工具源代码进行编译，生成容器镜像文件。

2. 应对工具成果包进行签名并验证机制，确保工具成果包的完整性和来源可信。

工具成果包签名验证流程如下：

a) 工具提供方准备好工具成果包；

b) 工具提供方使用哈希算法对工具成果包进行处理，生成该工具成果包的哈希值；

c) 工具提供方设置私钥对上一步生成的哈希值进行加密，生成数字签名；

d) 工具提供方将生成的数字签名附加到原始工具成果包中；

e) 工具使用方在接收到带有签名的成果包后，使用工具成果包内元数据描述文档中的公钥来解密数字签名，从而得到原始的哈希值；然后工具使用方再次使用相同的哈希算法对工具成果包本身进行处理，以获取另一个哈希值，并与原始的哈希值进行比对。

6 Web 页面工具开发要求

6.1 技术要求

6.1.1 浏览器兼容要求

Web 页面工具应兼容主流的国产化操作系统和浏览器。

6.1.2 字符编码

请求及返回结果应使用 UTF-8 编码。

6.1.3 协议类型

Web 页面工具应支持 HTTPS 协议。

6.1.4 URL 命名规范

Web 页面 URL 中的请求路径命名建议遵循如下规则：

1. 在所有 URL 命名中，应使用标准的英文单词或缩写，勿使用数字或特殊字符；
2. 所有命名都应遵循达意原则，即名称应含义清晰、明确；
3. 所有命名都不宜过长，应控制在 200 字符以内；
4. 命名采取**/{模块}/{功能定义}，建议采用动名词，能较为精准的反映真实含义。

6.1.5 HTML 规范

前端页面开发应符合 HTML5 规范，JS 适配 ECMAScript 6 版本。

6.2 前端页面开发要求

6.2.1 界面设计规范

Web 页面工具整体风格需要和平台保持一致，包括分辨率、页面布局和工具内部主色调等方面。

6.2.1.1 分辨率

工具页面应自适应屏幕分辨率，推荐分辨率为 1920px*1080px。

6.2.1.2 色调

工具在界面设计方面应采用主色调和辅助色，共同构成网页的标准色彩。主色调用来渲染界面重要或关键部分的颜色，辅助色用来支持主色调、融合主色调。比如工具标题栏的背景色一般采用主色调，正文内容采用辅助色。为保障工具的灵活性，建议将主要颜色设置在配置文件中，支持动态调整。

6.2.1.3 字体

工具的文字展示应统一进行设计和开发，包括字体、字号、间距等。将主要字体参数设置在配置文件中，便于灵活调整。

6.2.1.4 界面布局

Web 页面工具是集成到平台中进行使用，直接支撑构建业务场景。Web 页面工具应提供三种集成方式，第一种是将工具嵌入到场景页面中，第二种是在原页面弹出工具操作界面，第三种是弹出新网页来打开工具。三种方式分别对界面布局要求如下：

第一种方式要求工具界面直接展示内容；

第二种方式要求工具应采用上下布局，顶部是工具标题栏，用于说明工具的名称，提供关闭按钮，下方是工具本身具体内容。统一约定标题栏高度为 100px。

第三种方式要求工具应采用上下布局，顶部是工具标题栏，用于说明工具的名称，下方是工具本身具体内容。统一约定标题栏高度为 100px。

6.2.2 前端开发规范

Web 页面工具采用 iframe 方式嵌入场景或直接打页面的方式进行集成使用，工具开发时应考虑工具的独立性和易集成性。若工具涉及与场景深度交互操作，应参考以下技术要求：

6.2.2.1 输入规范

调用方通过工具页面组件输入信息，以 PostMessage 传递输入信息到工具页面组件，输入信息中需表明调用场景名称，输入参数示例如下：

表 6-1 页面工具输入参数表

名称	含义	规则说明
origin	表示哪个场景或系统在调用工具，此处填写场景或系统的规范名称的全称	必填
operation	需要工具执行什么操作	必填
input	传入工具的输入数据，由具体的工具来定义输入数据的组织结构，参考接口和参数设计规范，保持可读、易理解。	非必填

示例：

```

{
  "origin" : "XX 应用",
  "operation" : "draw",
  "input" : {
  }
}

```

6.2.2.2 输出规范

调用方根据工具页面传递的通信信息，调用相关功能进行响应，调用方传递数据或与工具页面通信时，需输入消息参数，并通过 PostMessage 传递消息至工具页面。消息参数属性如下：

表 6-2 消息参数表

名称	含义	规则说明
origin	消息来自哪个工具。此处填写工具的规范名称的全称，如：“遥感监测工具”	必填
operation	消息来自哪个操作	必填
result	工具返回的结果，result 内的内容由各个工具自行组织，应参考接口和参数设计规范	必填

示例：

```
{
  "origin" : "遥感监测工具",
  "operation" : "draw",
  "result" : {
  }
}
```

6.2.2.3 调用示例

在调用方系统中添加以下代码，使用 iframe 嵌入调用工具的页面组件，以调用遥感监测工具的页面组件为例：

```
<iframe src='/kjzlpvtv2/blockTool' id='ygjcTool'
name='ygjcTool'></iframe> 传值语法：
```

```
//监听遥感监测工具发来的信息，加载完成之后向遥感监测工具的页面组件输入
数据 window.addEventListener(
  'message' , handleMessage
)
function handleMessage() {
var ygjcWindow = document.getElementById('imageTool').contentwindow;
//传递数据到遥感监测页面组件
ygjcwindow.postMessage(message , "*");
}
```

6.3 后端服务开发要求

Web 页面工具涉及到业务逻辑需要调用后端 API 服务接口来处理，API 服务接口开发要求详见第 5 章 API 工具开发要求。

6.4 工具测试

1. 工具开发完成后，参照 6.1 技术要求对工具成果进行检查和测试，确保符合上述技术要求。

2. 对工具进行功能测试，确保工具能够正常使用，计算结果准确性达到预期。

3. 对工具进行并发测试，详细记录工具部署环境的资源配置，以及不同并发访问数下的响应情况。

4. 对工具前端 Web 页面进行兼容性测试，确保 Web 页面工具能够在主流的浏览器上正常运行，需要兼容的浏览器类型详见 6.1.1 浏览器兼容要求。

5. 对工具成果包进行安全扫描，在测试文档中记录扫描检查的内容项及检查结果，确保不存在已知安全漏洞。

6. 需要部署的工具在重新部署后，需根据工具使用手册说明，重新进行功能测试。

7. 整个工具测试完成后，需要将测试过程、测试结果及环境配置等信息记录到测试文档中，形成完整的测试报告。

6.5 工具编译与打包要求

1. Web 页面工具包括前端工程和后台服务两部分内容，在编译打包时需原则上尽量将两个程序整合到一起，生成一个容器镜像文件，从而提高工具部署的便捷性。对于复杂的工具，若前后端程序无法整合到一起，则应尽量减少独立部署的内容。

2. 应对工具成果包进行签名并验证机制，确保工具成果包的完整性和来源可信。工具签名及验证机制见 5.4 章节。

注：Web 页面工具开发要求同样适用于众筹场景技术要求。

7 SDK 工具开发要求

7.1 总体要求

SDK 工具开发，原则上不限定具体开发语言，但为了保证工具应用的广泛性，工具提供方应采用主流开发语言来完成工具开发，如 Java、C++、Python、JavaScript 等。

7.2 技术要求

7.2.1 类库设计规范

SDK 工具主要通过提供类库和方法来进行二次开发。为提高 SDK 工具使用便捷性，建议工具提供方参考以下规范：

1. 类库和方法命名可参照驼峰命名规则，与工具所处理的业务内容保持一致，方便开发人员查找和使用；

2. 参数定义要精炼和准确，参数不宜过多，只需暴露必要的参数，准确定义参数类型，特别是复合结构的数据，降低使用复杂度；

3. 返回结果应使用通用的 JSON 结构，尽量减少自定义结构体；

4. 应提供详细的操作日志，便于跟踪调试问题。

7.2.2 参数规范

避免使用硬编码的值，应通过参数传递必要的信息，并明确说明每个参数的名称、类型、是否必填等内容。推荐使用强类型参数，以减少类型错误的可能性。

7.2.3 调用规范

在调用 SDK 工具中的方法时，应对输入参数进行验证和校验，确保参数的有效性和安全性，避免传入无效的参数，影响工具的正常使用或影响计算结果的准确性。

7.2.4 错误处理和调试

在工具设计时，应统一考虑可能产生的异常或错误类别，设计一套完整的错误编码规范，以便开发者识别和处理问题。

7.3 SDK 内容要求

SDK 工具内容主要包括类库和方法，具体内容如下。

7.3.1 类库

工具提供方应提供类库清单，按照下表对每个类库进行详细描述，包括类名称、所属模块、主要功能和依赖说明，便于开发人员了解类库关系及用途。

表 7-1 类库清单表

类名称	所属模块	主要功能	依赖说明

7.3.2 方法

工具提供方应按照类库清单，将每个类库中的公共方法梳理出来，按照下表进行详细描述，包括方法名称、所属类库、功能描述、参数说明和返回结果。

表 7-2 方法说明表

方法名称	所属类库	功能描述	参数说明	返回结果

7.4 工具测试

1. 工具开发完成后，参照 7.1 技术要求对工具成果进行检查和测试，确保符合上述技术要求。
2. 对工具进行功能测试，确保工具能够正常使用，计算结果准确性达到预期。
3. 对工具成果包进行安全扫描，在测试文档中记录扫描检查的内容项及检查结果，确保不存在已知安全漏洞。
4. 整个工具测试完成后，需要将测试过程、测试结果及环境配置等信息记录到测试文档中，形成完整的测试报告。

7.5 编译与打包要求

1. 根据 SDK 开发语言和运行的操作系统选择相应的打包方式。
2. 应将 SDK 工具运行时依赖的环境打包到部署包中，确保“开箱即用”。
3. 应对工具成果包进行签名并验证机制，确保工具成果包的完整性和来源可信。

工具签名及验证机制见 5.4 章节。

8 工具安全要求

8.1 技术框架安全

工具开发不得使用含已知高危漏洞的组件框架。如后续出现安全漏洞，工具提供方应迅速对已用组件进行升级或替换，杜绝此类安全风险。

8.2 代码安全

1. 需要对输入的关键参数做合法性验证，包括但不限于验证数据的类型、长度、格式和范围。

2. 对特殊字符和潜在的危险输入/输出进行过滤或转义，以防止 SQL 注入、XML 注入、XML 外部实体注入、SSRF、命令注入、CRLF 漏洞、跨站脚本攻击（XSS）等安全漏洞。

3. 文件上传需增加特殊防护，包括但不限于后缀白名单、文件重命名、文件大小检测、文件类型检测、文件存储到不可解析的目录。

8.3 数据安全

1. 对工具关联的数据，根据数据敏感性和业务重要性进行数据分类和标签管理，为不同类别数据设定相应的保护级别和访问控制策略。

2. 对存储在数据库、文件系统、网络传输中的敏感数据进行加密处理，采用行业认可的加密算法（如 SM2、SM4）和密钥管理方案。

3. 对非必要的敏感数据进行脱敏处理（如去标识化、匿名化），在保证业务分析需求的同时降低数据泄露风险。

8.4 使用安全

1. 工具需要通过授权才能被用户使用，根据工具的不同类型，分别采用适合的授权方式。如 Web 页面工具和 API 工具宜采用 Token 码方式进行授权；SDK 工具宜采用 License 许可文件进行授权。

2. 工具涉及不同的数据和业务，需要采取权限管控措施，对用户分级分类，根据

不同的角色分配不同的权限。

8.5 日志与审计

工具需具备日志审计规范，记录的操作日志至少包括时间、地点、人物、事件、行为、操作对象等信息。

9 工具发布与共享要求

9.1 工具成果包

工具成果包应包含工具基本信息、数据要求、相关手册、使用说明、部署包等内容。工具打包后形成一个完整的 zip 包，结构如下图所示：

```
|-----***工具成果包.zip
|-----01 工具白皮书.doc
|-----02 数据要求
|-----数据清单.doc
|-----测试数据.zip
|-----03 手册类
|-----工具安装部署手册.pdf
|-----工具操作使用手册.pdf
|-----测试报告.pdf
|-----04 使用说明
|-----工具参数说明及示例代码.md
|-----演示效果视频.mp4
|-----05 部署包
|-----tools.tar.gz/tools.jar/.....
|-----metadata.xml
```

图 9-1 工具成果包结构

1. 工具白皮书文档用来描述工具的名称、简介、应用场景、工具建设等基础信息。
2. 数据要求目录包含存放工具所需要的数据清单及结构要求，详细要求见 5.3 章节，同时提供一份测试数据。
3. 手册类目录存放工具相关文档成果，包括工具安装部署手册、操作使用手册以及测试报告等。
4. 使用说明目录存放工具参数说明和调用示例代码的文档以及演示效果视频，工具参数说明及示例代码包括 API 接口列表、参数定义、调用示例等内容，以 markdown 文档格式编写。应用案例视频需以 H264 编码的 MP4 格式提供，文件大小不超过 500MB，以及不少于 3 张应用效果图。
5. 部署包目录存放工具按照技术要求编译打包生成后的成果，如 Docker 镜像或 Jar 包等；同时，包括工具成果包的元数据说明文档，包括工具的注册信息、密钥及签名信息等。

9.2 工具发布流程

由工具提供方准备好工具成果包内容，在平台工具中心进行注册，经过工具管理

方审核通过后进行发布。工具发布流程如下图所示：

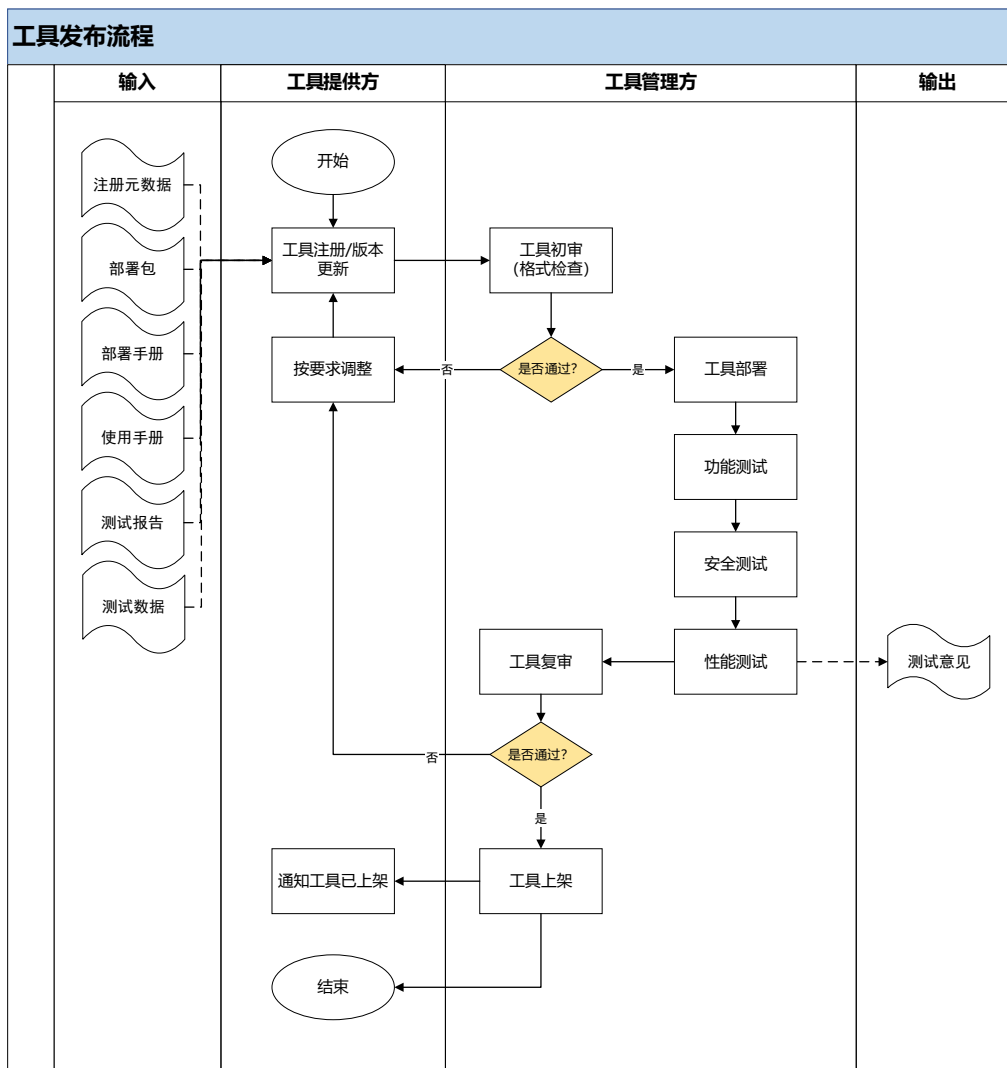


图 9-2 工具发布流程图

工具提供方和工具管理方参与工具发布流程，各方具体工作说明如下：

9.2.1 工具提供方

1. 按 9.1 要求准备好工具部署包，登录平台工具中心进行工具注册或者版本更新；
2. 若注册工具审查不通过，需要按照反馈的意见进行调整，然后重新提交。

9.2.2 工具管理方

1. 对工具提供方提交的工具信息进行初步审查，若审查不通过，需反馈审查意见至工具提供方；
2. 对工具进行本地化部署和测试，按照工具部署说明在独立的环境下重新工具，从功能、性能和安全等多方面进行测试验证，并出具测试意见；
3. 根据测试意见对工具进行复审；若复审不通过，需反馈复审意见至工具提供方；

复审通过后，直接将其发布到工具中心进行上架使用，并发送通知给到工具提供方。

9.3 工具共享流程

按照“谁提供、谁审核、谁授权”原则进行工具的申请与审核。工具共享流程如下图所示：

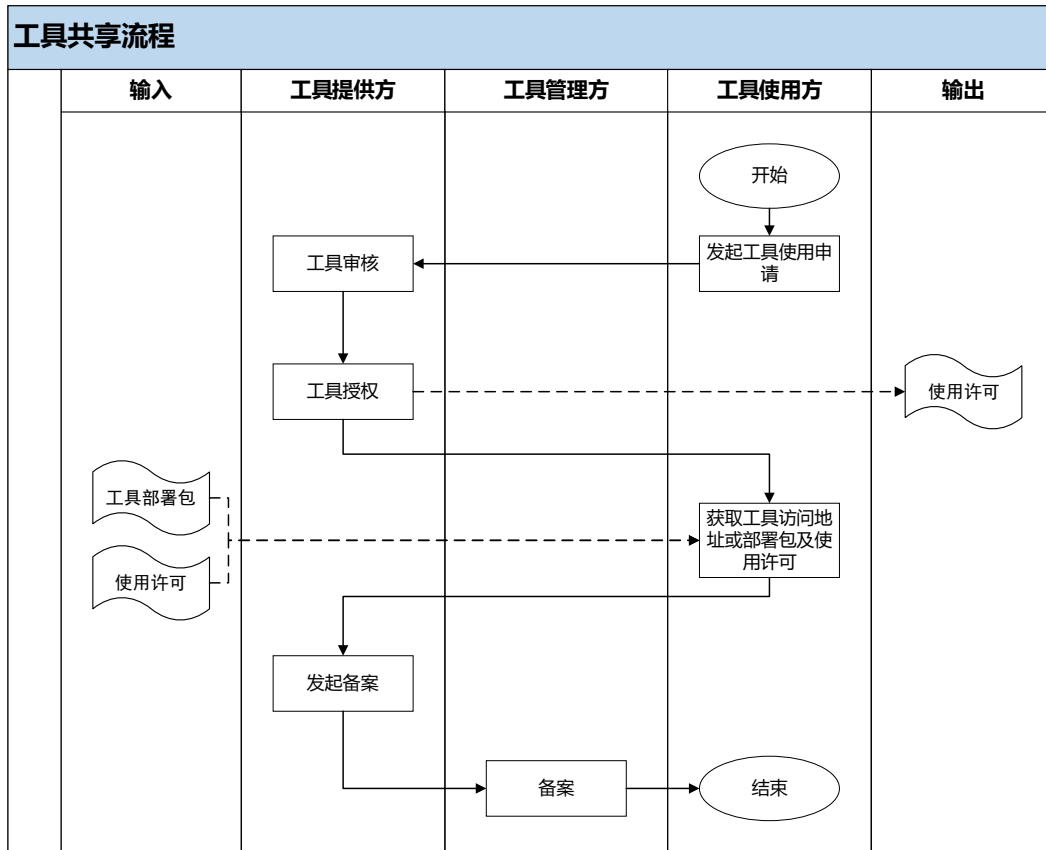


图 9-3 工具共享流程图

工具使用方、工具提供方和工具管理方参与工具发布流程，各方具体工作说明如下：

9.3.1 工具使用方

1. 从平台工具中心查找目标工具，在线提交使用申请；
2. 与工具提供方沟通确认后，获取工具访问地址或部署包以及使用许可信息进行本地化应用；在实际应用工具后，对工具使用情况进行评价反馈。

9.3.2 工具提供方

1. 对工具使用方提出的申请进行审核，确认后进行工具授权；
2. 将工具访问信息或部署包以及使用许可推送给工具提供方；
3. 将工具共享涉及相关信息发起备案，提交给工具管理方。

9.3.3 工具管理方

管理工具共享应用备案信息。

9.4 附表 国土空间基础信息平台工具注册表

(一) 工具基本信息			
*工具名称	命名简明扼要，原则不超过 15 个汉字。		
*工具简介	填写工具简要介绍（200 汉字以内）		
建设依据	描述该工具建设时参考的依据，如部里发文、标准规范，或参考的论文等。		
应用场景	描述该工具适用的应用场景。参考场景：耕地保护、生态保护、防灾减灾、矿产资源安全、历史文化保护；自然资源调查监测、国土空间规划实施监测、研讨会修复；要素配置、市场监管、节约集约；确权登记与不动产登记、资产权益、督察执法、行业监管、政务管理、社会服务。可填写多个。对于基础算法/模型工具，可以不填。		
*适用范围	适用范围填写要求：描述工具适用的地域范围：0-不限 1-部级 2-省级 3-市级 4-县级 5-特定 6-其他。	*所属分类	包括通用工具和专用工具，通用工具包括基础算法/模型工具；数据处理工具；空间分析工具；其他。专用工具包括自然资源要素监测识别；国土空间智慧规划和治理；智能审查审批；公共服务；决策支持；其他。
*版本号	V 主版本号. 子版本号[. 修正版本号]	*工具图标	作为工具的标识，提供大小 72*72，正方形 png 图片作为图标
*工具类型	指该工具技术分类。1-API 工具：提供服务地址、开发人员以在线调用的方式进行使用。2-Web 页面工具：提供可操作的页面，业务系统可直接将该页面嵌入进来直接操作使用。3-SDK 工具：提供二次开发包，开发人员在代码框架中引入该依赖进行二次开发。	*应用类型	指该工具应用类型。1-必备工具：指要求各级自然资源管理部门必须使用此工具。一般是指按照国家发布的强制性标准进行开发的工具，如数据质检工具、业务指标审查工具等。2-推荐使用：指具有一定先进特点的工具，如智能化工具、业务常用工具或计算效率高等特点的工具，可帮助用户大幅提升工作效率。3-选择性使用：指普通工具，用户根据各自需要选择使用。

*应用方式	指工具应用方式。1-在线应用：指通过服务地址直接在线访问智能工具的方式，要求调用方与智能工具部署在同一个网络环境下。2-下载应用：指工具使用方将该工具部署包下载到本服务器环境进行安装部署，供本地业务系统调用。	*工具类别	包括原创工具和复合工具两类。1-原创工具：指工具提供方发布的原子级的工具；2-复合工具：指在已有的工具基础上进行扩展升级而发布的新版本，或者将多个已有工具进行整合而产生的新的工具。
部署位置	针对在线使用的工具部署运行的位置。1-统一部署：指将各方众筹共享出来的工具在自然资源部云环境中进行集中部署，提供在线服务的方式供各类用户调用。2-提供方部署：指工具提供方在本地环境部署好该工具，将访问工具的服务地址等信息注册到管理中心，供各类用户调用。3-使用方部署：指使用方从工具管理中心下载智能工具的部署包，然后在本地服务器环境部署该工具，供本地用户使用。	是否为智能工具	描述该工具是否为智能工具，1-是智能工具；2-非智能工具。
关键词	给工具打标签；类似论文的关键词，可填写多个。		
(二) 工具生产方信息			
*工具提供单位名称	指工具提供方的单位全称（特指自然资源信息化建设单位或科研院所单位名称）。		
*工具提供单位联系方式	填写工具提供方的联系人、联系电话和联系邮箱等信息。		
需求提出单位名称	如果该工具有需求方，填写提出建设该工具的单位名称。（用于众筹需求由部或地方单位提出）		
需求提出单位联系方式	填写工具需求提供方的联系人、联系电话和联系邮箱等信息。		
工具研发单位名称	实际完成该工具研发的单位名称。如果工具提供单位即研发单位，可不填。		
工具研发单位联系方式	填写工具研发单位的联系人、联系电话和联系邮箱等信息。		
(三) 工具详细信息			
*工具主要功能描述	描述工具实现的主要功能以及工具的输入输出要求等。（500字以内）		

工具主要特色或创新点	总结工具主要技术、功能、应用等方面的亮点或创新点。(200字以内)		
工具应用案例描述	如果工具已经投入应用, 简要介绍工具应用案例, 以及工具在应用中所起的作用。(500字以内)		
工具访问信息	按照工具调用方式的不同, 在填写工具访问信息时要求不同, 参照不同类开的工具技术要求。		
*工具开发环境说明	介绍工具开发环境, 包括操作系统、开发语言/引擎、数据库和版本、GIS环境和版本、第三方软件等情况(200字以内)		
*工具技术构架	填写工具软件开发技术构架。包括: C/S架构、B/S架构、SOA架构、微服务架构等。		
工具运行环境要求	针对应用方式为“离线应用”的工具。介绍工具运行时所必须的操作系统、中间件、GIS软件、数据库等软件环境, 以及对服务器配置要求等(500字以内)		
是否依赖商业数据库	针对应用方式为“离线应用”的工具。该工具运行是否依赖商业数据库软件。1-依赖; 2-不依赖。	商业数据库信息	针对应用方式为“离线应用”的工具。如果依赖商业数据库软件, 填写商业数据库名称、类型和版本号。(50字以内)
是否依赖商业GIS软件	针对应用方式为“离线应用”的工具。该工具运行是否依赖商业GIS软件。1-依赖; 2-不依赖。	商业GIS软件信息	针对应用方式为“离线应用”的工具。如果依赖商业GIS软件, 填写商业GIS软件名称、类型和版本号。(50字以内)
依赖其他软件说明	针对应用方式为“离线应用”的工具。如果依赖其他第三方软件, 填写软件名称等信息。(50字以内)		
*是否支持信创环境	该工具运行是否支持信创环境。1-支持; 2-不支持。	信创环境说明	如果支持信创环境, 填写支持的信创环境(客户端、数据库等)(100字以内)
其他说明	填写工具其他需要说明的情况。(500字以内)		
(四) 附件文件			
*工具应用效果视频或图片	以附件方式上传工具应用效果视频(1个文件, MP4, 大小不超过500M, 时长不超过5分钟), 或者图片(可多张, JPG或GIF格式, 每张大小不超过20M)		
*工具操作手册	提供PDF格式文件。详细介绍工具使用方法。对于应用方式为“在线应用”的工具, 介绍工具操作方法和步骤; 对于应用方式为“离线应用”的工具, 详细介绍输入参数和输出结果, 需给出调用示例, 如果工具有数据(含空间)要求, 需要详细介绍工具运行时所需要的数据清单, 包括数据分类、数据项、数据项说明、数据精度、数据用途、数据表、数据源部门的信息等以及数据处理工具等要求。		
*工具部署手册	提供PDF格式文件。详细介绍工具部署方法。包括工具所需软硬件环境及工具软件部署操作步骤及说明。		

*工具测试数据	提供测试数据，支撑工具本地化部署与测试。与工具输入数据格式要求保持一致。
*工具测试报告	提供 PDF 格式文件。详细介绍工具测试步骤、测试过程、测试结论等内容。
工具部署包	除应用方式为“在线应用”且部署位置为“提供方部署”的工具，均需提供经测试的完整部署包文件。
注：表中所有带*表示必填项。	