

Automated Testing Interface (Inter USS)

Rotas que devem ser implementadas pelo provedor para poder realizar os testes automatizados implementados pelo USS Qualifier.

- [Remote ID Test Data Injection \(v0.5.1\)](#)
- [Remote ID Display Data Observation \(v0.3.0\)](#)
- [Flight Planning Automated Testing Interface \(v0.7.0\)](#)
- [Geospatial Map Provider Automated Testing Interface \(v0.2.3\)](#)
- [Geo-Awareness Automated Test Interfaces \(v0.2.2\)](#)

Remote ID Test Data Injection (v0.5.1)

Esta interface é fornecida por cada Provedor de Serviço (Service Provider) que deseja ser testado pelo framework de testes automatizados. A suíte de testes chama esta interface para injetar dados de voo no sistema sob teste.

Segurança (Autenticação)

- **Tipo:** OAuth2 (Client Credentials)
- **Escopo Necessário:** rid.inject_test_data
- **Descrição:** Acesso concedido para injetar dados de teste em um Provedor de Serviços.

Endpoints

1. Criar Teste

Rota: `PUT /tests/{test_id}`

Solicita a criação de um ou mais voos lógicos baseados na injeção dos dados fornecidos. O ID de injeção (injection_id) não pode ser modificado pelo provedor.

- **Parâmetros de Rota:** test_id (obrigatório) - O ID do teste. Exemplo: 2979bd18-7f06-441c-bda6-e82c841c35d6
- **Corpo da Requisição:** Requer o objeto CreateTestParameters (uma lista de voos).
- **Respostas Esperadas:**
 - **200 OK:** Teste criado com sucesso. Retorna ChangeTestResponse.
 - **409 Conflict:** O teste já existe.

2. Deletar Teste

Rota: `DELETE /tests/{test_id}/{version}`

Remove todos os dados de teste associados a este teste do injetor do Provedor de Serviços.

- **Parâmetros de Rota:** test_id (obrigatório) e version (obrigatório).
- **Respostas Esperadas:**
 - **200 OK:** Teste deletado com sucesso. Retorna DeleteTestResponse.

3. Consultar Notificações de Usuário

Rota: GET /user_notifications

Retorna a lista de notificações observadas pelo usuário virtual. As notificações devem estar disponíveis em até 5 segundos após a observação.

- **Parâmetros de Query:** * after (obrigatório): Não incluir notificações observadas antes deste horário.
 - before (opcional): Não incluir notificações após este horário. O padrão é o agora.
- **Respostas Esperadas:**
 - **200 OK:** Notificações recuperadas com sucesso.
 - **400 Bad Request:** Faltando o parâmetro 'after' ou requisição inválida.
 - **401 Unauthorized:** Token ausente, não pôde ser decodificado ou é inválido.
 - **403 Forbidden:** Token decodificado, mas sem o escopo apropriado.

Payload de Exemplo (JSON)

Formato esperado para o envio das informações do voo:

JSON



```
{
  "requested_flights": [
    {
      "injection_id": "edb7695f-8737-4b9f-91f8-e2afbb333f41",
      "aircraft_type": "Aeroplane",
      "telemetry": [
        {
          "timestamp": "2024-04-22T16:36:50.52Z",
          "position": {
            "lat": -23.1791,
            "lng": -45.8872,
```

```
    "alt": 100.0
  }
}
],
"details_responses": [
  {
    "effective_after": "2024-04-22T16:36:50.52Z",
    "details": {
      "id": "a3423b-213401-0023"
    }
  }
]
}
]
```

Remote ID Display Data Observation (v0.3.0)

Esta interface é fornecida por cada Provedor de Exibição (Display Provider) que deseja ser testado pelo framework de testes automatizados. A suíte de testes chama esta interface para obter as informações atuais do Remote ID sob a perspectiva de um usuário do Provedor de Exibição.

Segurança (Autenticação)

- **Tipo:** OAuth2 (Client Credentials)
- **Escopo Necessário:** dss.read.identification_service_areas
- **Descrição:** Acesso concedido para ler informações atuais do Remote ID. O token JWT deve ser enviado no header `Authorization: Bearer <token>`.

Endpoints

1. Consultar Dados de Exibição (Poll Display Data)

Rota: `GET /display_data`

Solicita os dados atuais de exibição do Remote ID da mesma forma que seriam visualizados por uma Aplicação de Exibição (Display Application).

- **Parâmetros de Query:** * view (obrigatório): A área desta visualização no formato lat1,lng1,lat2,lng2. A visualização é a menor caixa delimitada (bounding box) pelos pontos de canto fornecidos. Exemplo: 29.97816,31.13296,29.98025,31.13535
- **Respostas Esperadas:**
 - **200 OK:** Dados de exibição do Remote ID recuperados com sucesso. Retorna o schema `GetDisplayDataResponse` (contendo listas de voos e clusters).

2. Obter Detalhes do Voo

Rota: `GET /display_data/{id}`

Obtém os detalhes de um voo específico que foi previamente identificado através da rota `/display_data`.

- **Parâmetros de Rota:** * id (obrigatório): O identificador do voo. Exemplo: 1e3adb99-acc9-424f-a04e-a0743538849a
- **Respostas Esperadas:**
 - **200 OK:** Detalhes sobre o voo solicitado foram recuperados com sucesso. Retorna o schema `GetDetailsResponse`.
 - **404 Not Found:** O voo solicitado não foi encontrado.

Payloads de Exemplo (JSON)

Exemplo de Resposta para `/display_data`

Formato esperado retornando voos conhecidos e aglomerados (clusters) onde a posição precisa não é exata:

JSON

```
{
  "flights": [
    {
      "id": "1e3adb99-acc9-424f-a04e-a0743538849a",
      "aircraft_type": "Aeroplane",
      "current_state": {
        "timestamp": "2024-04-22T16:36:50.52Z",
        "operational_status": "Airborne"
      },
      "most_recent_position": {
        "lat": -23.1791,
        "lng": -45.8872,
        "alt": 100.0
      }
    }
  ],
  "clusters": [
```

```
{
  "corners": [
    { "lat": -23.179, "lng": -45.887 },
    { "lat": -23.180, "lng": -45.888 }
  ],
  "area_sqm": 15000.5,
  "number_of_flights": 3
}
]
```

Exemplo de Resposta para

`/display_data/{id}`

Formato esperado retornando os detalhes do operador e da aeronave:

JSON

```
{
  "operator": {
    "id": "OP-BR-987654321",
    "location": {
      "lat": -23.1805,
      "lng": -45.8881
    },
    "altitude": {
      "altitude": 550.0,
      "altitude_type": "Takeoff"
    }
  },
  "uas": {
    "id": "UAS-XT-550",
    "eu_classification": "Class0"
  }
}
```


Flight Planning Automated Testing Interface (v0.7.0)

Esta interface é fornecida por um USS (UAS Service Supplier) que deseja participar de testes automatizados envolvendo tentativas de planejamento de voo. Um cliente (geralmente o `uss_qualifier`) instrui um usuário virtual a interagir com a interface do USS para planejar, atualizar e fechar planos de voo.

Segurança (Autenticação)

- **Tipo:** OAuth2 (Client Credentials)
- **Escopos Necessários:**
 - `interuss.flight_planning.direct_automated_test`: Permite determinar a prontidão do USS para o teste e instruir o administrador a preparar a área de testes.
 - `interuss.flight_planning.plan`: Permite enviar instruções ao usuário virtual para planejar, modificar ou fechar um plano de voo.
- **Descrição:** O token JWT deve ser enviado no header `Authorization: Bearer <token>`.

Endpoints

1. Consultar Status da Interface

Rota: `GET /status`

Obtém o status atual desta interface de testes automatizados.

- **Escopo Exigido:** `interuss.flight_planning.direct_automated_test`
- **Respostas Esperadas:**
 - **200 OK:** Interface disponível. Retorna se está `Starting` ou `Ready`.
 - **404 Not Found:** A interface de testes não está disponível.

2. Limpar Área (Clear Area)

Rota: `POST /clear_area_requests`

Solicita que o administrador do USS cancele e remova todos os planos de voo gerenciados por ele que interceptem a área especificada no payload.

- **Escopo Exigido:** `interuss.flight_planning.direct_automated_test`
- **Corpo da Requisição:** Requer `request_id` único e a extensão da área (`extent`).
- **Respostas Esperadas:**
 - **200 OK:** Área limpa com sucesso. Retorna `ClearAreaResponse` (indicando sucesso ou detalhes de falha na limpeza).

3. Criar ou Atualizar Plano de Voo (Upsert)

Rota: `PUT /flight_plans/{flight_plan_id}`

Simula a intenção de um usuário de enviar um plano de voo novo ou atualizado.

- **Escopo Exigido:** `interuss.flight_planning.plan`
- **Parâmetros de Rota:** `flight_plan_id` (obrigatório) - UUID formato v4. Ex: `03e5572a-f733-49af-bc14-8a18bd53ee39`
- **Corpo da Requisição:** Requer o objeto `UpsertFlightPlanRequest`.
- **Respostas Esperadas:**
 - **200 OK:** Dados processados com sucesso. Retorna `UpsertFlightPlanResponse` com o status do planejamento (`Completed`, `Rejected`, `Failed`, etc).
 - **409 Conflict:** Conflito de `request_id` duplicado ou outra condição de conflito.

4. Fechar/Deletar Plano de Voo

Rota: `DELETE /flight_plans/{flight_plan_id}`

Permite que o diretor de testes instrua o USS a remover um plano de voo que não é mais necessário para os testes.

- **Escopo Exigido:** `interuss.flight_planning.direct_automated_test`
- **Parâmetros de Rota:** `flight_plan_id` (obrigatório).
- **Respostas Esperadas:**
 - **200 OK:** Plano de voo deletado com sucesso. Retorna `DeleteFlightPlanResponse`.
 - **404 Not Found:** Plano de voo não encontrado (já pode ter sido deletado).

5. Consultar Notificações de Usuário

Rota: `GET /user_notifications`

Retorna a lista de notificações observadas pelo usuário virtual. Devem estar disponíveis para consulta em até 5 segundos após a observação.

- **Escopo Exigido:** `interuss.flight_planning.plan`
- **Parâmetros de Query:** * `after` (obrigatório): Limite inferior de tempo.
 - `before` (opcional): Limite superior de tempo (padrão é o momento atual).
- **Respostas Esperadas:**
 - **200 OK:** Notificações recuperadas.
 - **400 Bad Request:** Parâmetros de tempo ausentes ou inválidos.

Payload de Exemplo (JSON)

Formato simplificado esperado para a criação/atualização de um plano de voo (`PUT /flight_plans/{flight_plan_id}`):

JSON



```
{
  "request_id": "b5a9b837-1234-4a2b-9876-c5096a295a12",
  "execution_style": "IfAllowed",
  "flight_plan": {
    "basic_information": {
      "usage_state": "Planned",
      "description": "Medical supplies delivery operated by Example Drone Company",
      "utm_id": "ae1fa066-6d68-4018-8274-af867966978e",
      "area": [
        {
          "volume": {
            "outline_polygon": {
              "vertices": [
                { "lat": -23.179, "lng": -45.887 },
                { "lat": -23.180, "lng": -45.888 }
              ]
            },
            "altitude_lower": { "value": 0, "reference": "W84", "units": "M" },
            "altitude_upper": { "value": 120, "reference": "W84", "units": "M" }
          },
          "time_start": { "value": "2024-04-22T16:30:00Z", "format": "RFC3339" },
          "time_end": { "value": "2024-04-22T17:30:00Z", "format": "RFC3339" }
        }
      ]
    }
  }
}
```

```
}  
]  
}  
}  
}
```

Geospatial Map Provider Automated Testing Interface (v0.2.3)

Esta interface é implementada para fornecer ao `uss_qualifier` (e possivelmente a outros clientes semelhantes) a capacidade de simular um usuário interagindo com o mapa geoespacial do USS (UAS Service Supplier) para avaliar potenciais planos de voo.

A suíte de testes automatizados pode chamar esta interface para carregar dados de teste no USS e, em seguida, realizar consultas virtuais no mapa geoespacial para avaliar o processamento e a interpretação de dados de *Geo-Awareness*.

Segurança (Autenticação)

- **Tipo:** OAuth2 (Client Credentials)
- **Escopos Necessários:**
 - `interuss.geospatial_map.direct_automated_test`: Permite determinar a prontidão do USS para teste e instruir o administrador a carregar fontes de dados geoespaciais (agindo como diretor de testes).
 - `interuss.geospatial_map.query`: Permite ler informações do mapa geoespacial do USS como seriam vistas por um usuário normal.
- **Descrição:** O token JWT deve ser enviado no header `Authorization: Bearer <token>`.

Endpoints

1. Consultar Status da Interface

Rota: `GET /status`

Obtém o status desta interface de testes automatizados.

- **Escopo Exigido:** `interuss.geospatial_map.direct_automated_test`

- **Respostas Esperadas:**

- **200 OK:** Interface disponível. Retorna `Starting` ou `Ready`.
- **404 Not Found:** A interface não está disponível.

2. Listar Fontes de Dados Geoespaciais

Rota: `GET /geospatial_data_sources`

Lista todas as fontes de dados geoespaciais carregadas pelo administrador do USS a pedido do diretor de testes.

- **Escopo Exigido:** `interuss.geospatial_map.direct_automated_test`
- **Respostas Esperadas:**
 - **200 OK:** Retorna o schema `ListGeospatialDataSourcesResponse` (lista de fontes e seus respectivos status).

3. Importar e Ativar Fonte de Dados

Rota: `PUT /geospatial_data_sources/{geospatial_data_source_id}`

Instrui o administrador do USS a importar e ativar dados geoespaciais da fonte especificada (como um link HTTPS para um JSON ED-269).

- **Parâmetros de Rota:** `geospatial_data_source_id` (obrigatório, formato UUID v4).
- **Corpo da Requisição:** `CreateGeospatialDataSourceRequest`.
- **Respostas Esperadas:**
 - **200 OK:** O USS foi instruído a importar e ativar os dados. Retorna `GeospatialDataSourceResponse`.
 - **409 Conflict:** Um pedido com este ID já foi feito para carregar uma fonte de dados diferente.

4. Consultar Status de uma Fonte de Dados

Rota: `GET /geospatial_data_sources/{geospatial_data_source_id}`

Obtém o status de processamento/ativação de uma fonte de dados específica no USS.

- **Respostas Esperadas:**
 - **200 OK:** Retorna status (`Activating`, `Ready`, `Error`, etc).
 - **404 Not Found:** A fonte de dados foi desativada com sucesso ou não existe.

5. Desativar Fonte de Dados

Rota: DELETE /geospatial_data_sources/{geospatial_data_source_id}

Instrui o administrador do USS a desativar e excluir a fonte de dados geoespaciais.

- **Respostas Esperadas:**

- **200 OK:** Requisição processada. O status da deleção pode ser checado via GET (até retornar 404).

6. Consultar Mapa Geoespacial (Queries)

Rota: POST /map/queries

Verifica se uma ou múltiplas características geoespaciais (Geozones/Features) se aplicam a uma posição de interesse para um período de tempo especificado ou condições operacionais (Simulação de clique/consulta no mapa).

- **Escopo Exigido:** interuss.geospatial_map.query
- **Corpo da Requisição:** GeospatialMapQueryRequest (contendo filtros de volume 4D, tempo, regra de operação, etc).
- **Respostas Esperadas:**
 - **200 OK:** Consulta realizada. Retorna se há intersecção/presença de características geoespaciais (schema GeospatialMapQueryReply com status Present, Absent, etc).

Payloads de Exemplo (JSON)

Exemplo de Criação de Fonte de Dados (PUT)

JSON

```
{
  "https_source": {
    "url": "https://caa.example.com/geozones.json",
    "format": "ED-269"
  }
}
```

```
}  
}
```

Exemplo de Resposta de Status da Fonte

JSON

```
{  
  "data_source": {  
    "id": "32ef0162-30c4-4e56-9ce0-b204155ef93d",  
    "status": "Ready"  
  }  
}
```

Exemplo de Consulta no Mapa (POST /map/queries)

JSON

```
{  
  "checks": [  
    {  
      "filter_sets": [  
        {  
          "position": {  
            "location": { "lat": -23.179, "lng": -45.887 },  
            "altitude": { "value": 100, "reference": "W84", "units": "M" }  
          },  
          "operation_rule_set": "Part107",  
          "resulting_operational_impact": "BlockOrAdvise",  
        }  
      ]  
    }  
  ]  
}
```

```
"ed269": {  
  "u_space_class": "EUROCONTROL"  
}  
]  
]  
]  
]  
]
```

Geo-Awareness Automated Test Interfaces (v0.2.2)

Esta interface é implementada por cada USS (UAS Service Supplier) que deseja ser testado pelo framework de testes automatizados. A suíte de testes chama esta interface para carregar dados de teste no sistema do USS sob teste e ler informações de *Geo-Awareness* (Consciência Geográfica) para avaliar seu processamento e interpretação.

Segurança (Autenticação)

- **Tipo:** OAuth2 (Client Credentials)
- **Escopo Necessário:** `geo-awareness.test`
- **Descrição:** Permite que o cliente instrua o USS sob teste a carregar dados de *Geozone* e ler informações de *Geo-Awareness*. O token JWT deve ser enviado no header `Authorization: Bearer <token>`.

Endpoints

1. Consultar Status da Interface

Rota: `GET /status`

Obtém o status atual da interface de testes automatizados do USS.

- **Escopo Exigido:** `geo-awareness.test`
- **Respostas Esperadas:**
 - **200 OK:** Interface ativada. Retorna se está `Starting` (iniciando) ou `Ready` (pronta para receber testes).
 - **404 Not Found:** A interface de testes não está ativada.

2. Importar e Ativar Fonte de Geozone

Rota: `PUT /geozone_sources/{geozone_source_id}`

Instrui o USS a importar e ativar dados de *Geozone* a partir de uma fonte especificada (como um link HTTPS para um JSON).

- **Parâmetros de Rota:** `geozone_source_id` (obrigatório) - Um UUID v4 que identifica a fonte de dados.
- **Corpo da Requisição:** Requer o objeto `CreateGeozoneSourceRequest`.
- **Respostas Esperadas:**
 - **200 OK:** A requisição foi processada e o USS foi instruído a importar os dados. Retorna o schema `GeozoneSourceResponse`.

3. Consultar Status da Fonte de Geozone

Rota: `GET /geozone_sources/{geozone_source_id}`

Obtém o status de processamento da fonte de *Geozone* e de seus dados dentro do USS.

- **Respostas Esperadas:**
 - **200 OK:** A fonte existe. Retorna o status atual (`Activating`, `Ready`, `Deactivating`, `Unsupported`, `Rejected`, `Error`). Em caso de erro, uma mensagem de *debug* é incluída.
 - **404 Not Found:** A fonte de *Geozone* não existe ou já foi desativada com sucesso.

4. Desativar Fonte de Geozone

Rota: `DELETE /geozone_sources/{geozone_source_id}`

Instrui o USS a desativar e excluir a fonte de *Geozone* e todos os seus dados.

- **Respostas Esperadas:**
 - **200 OK:** Requisição processada e os dados serão deletados. O status da deleção deve ser verificado usando o método GET até que ele retorne 404.

5. Checar Geozones Aplicáveis (Check)

Rota: `POST /geozones/check`

Verifica se uma ou múltiplas *Geozones* são aplicáveis a uma posição de interesse, para um período de tempo e condições operacionais especificadas.

- **Corpo da Requisição:** `GeozonesCheckRequest` (lista de checagens contendo conjuntos de filtros).
- **Respostas Esperadas:**

- **200 OK:** A checagem foi realizada com sucesso. Retorna `GeozonesCheckReply`, indicando se a *Geozone* está `Present`, `Absent`, `UnsupportedFilter` ou `Error` para cada filtro solicitado.

Payloads de Exemplo (JSON)

Exemplo de Importação de Fonte (PUT)

JSON

```
{
  "https_source": {
    "url": "https://caa.example.com/geozones.json",
    "format": "ED-269"
  }
}
```

Exemplo de Resposta de Status (GET `/geozone_sources/...`)

JSON

```
{
  "result": "Ready"
}
```

Exemplo de Checagem de Geozones (POST /geozones/check)

Formato esperado para simular a validação de uma posição contra o banco de dados de zonas:

JSON

```
{
  "checks": [
    {
      "filterSets": [
        {
          "position": {
            "uomDimensions": "M",
            "verticalReferenceType": "AGL",
            "height": 120.0,
            "longitude": -45.8872,
            "latitude": -23.1791
          },
          "after": "2024-04-22T16:30:00Z",
          "before": "2024-04-22T18:30:00Z",
          "ed269": {
            "uSpaceClass": "EUROCONTROL",
            "acceptableRestrictions": ["PROHIBITED", "REQ_AUTHORISATION"]
          }
        }
      ]
    }
  ]
}
```