



Overeenkomst balanspartner

Rijkswaterstaat Windpark Maasvlakte 2

RWS BEDRIJFSVERTROUWELIJK

BIJLAGE 6 Raakvlak Windpark Maasvlakte 2

Datum	27 oktober 2021
Status	Concept
Versie	A
Zaaknummer	31172943

Inhoud

1. Ontwerp & implementatie van Technische Automatisering en de interface met de balanspartner.....	3
1.1 Ontsluiting van het windpark: de technische verbinding.....	3
1.2 Communicatie van geplande (on)beschikbaarheid.....	3
Annex 1: IT Overzicht.....	4
Annex 2: Wind turbine availability API manual V 2.3.....	5

1. ONTWERP & IMPLEMENTATIE VAN TECHNISCHE AUTOMATISERING EN DE INTERFACE MET DE BALANSPARTNER

1.1 Ontsluiting van het windpark: de technische verbinding

Eneco hanteert voor het ontsluiten van haar windparken een aantal standaardoplossingen voor de netwerkverbinding en achterliggende infrastructuur. Deze staan in onderstaande tabel weergegeven. Voor het windpark Maasvlakte II zal Eneco de High-End oplossing implementeren, zodat maximale beschikbaarheid en betrouwbaarheid van de verbinding met het park wordt bereikt.

	HIGH-END
VERBINDING	Vaste verbinding
BACKUP (REDUNDANTE) VERBINDING	4G
LOKALE NETWERK INFRASTRUCTUUR BIJ DE ASSET	Firewalls, switches, routers

Naast de netwerkinfrastructuur is hardware nodig om monitoring & aansturing van het windpark mogelijk te maken. Bij windparken van Eneco is het standaard gebruik dat de balanspartner en Asset Management ieder hun eigen hardware verzorgen en in beheer nemen.

Om monitoring & curtailment door de balanspartner te faciliteren vraagt Eneco van de balanspartner het volgende:

- (a) De balanspartner maakt gebruik van de verbinding en lokale netwerkinfrastructuur van Eneco. De balanspartner legt dus geen eigen verbinding aan. Eneco is verantwoordelijk voor beheer en onderhoud van de verbinding en bijbehorende netwerkinfrastructuur.
- (b) De balanspartner ontwerpt en plaatst een eigen hardware oplossing t.b.v. monitoring & aansturing van het windpark door de balanspartner (curtailment). De balanspartner is verantwoordelijk voor beheer en onderhoud van de eigen hardware oplossing.

1.2 Communicatie van geplande (on)beschikbaarheid

Voor de communicatie van de geplande (on)beschikbaarheid vanuit het Operations & Control Center richting de balanspartner maakt Eneco gebruik van de Asset Availability Tool. De balanspartner kan met deze tool verbinding maken via een Rest API. De documentatie van deze API is beschikbaar.

In de Asset Availability tool wordt:

- (a) Geplande niet-beschikbaarheid (als gevolg van onderhouds- en servicewerkzaamheden door Vestas) bijgewerkt zodra nieuwe informatie beschikbaar is.
- (b) De laatste update van deze geplande niet-beschikbaarheid uiterlijk day-ahead om 9:00 beschikbaar gemaakt.
- (c) Na 9:00 uur day-ahead wordt geen nadere prognose over niet-beschikbaarheid afgegeven.

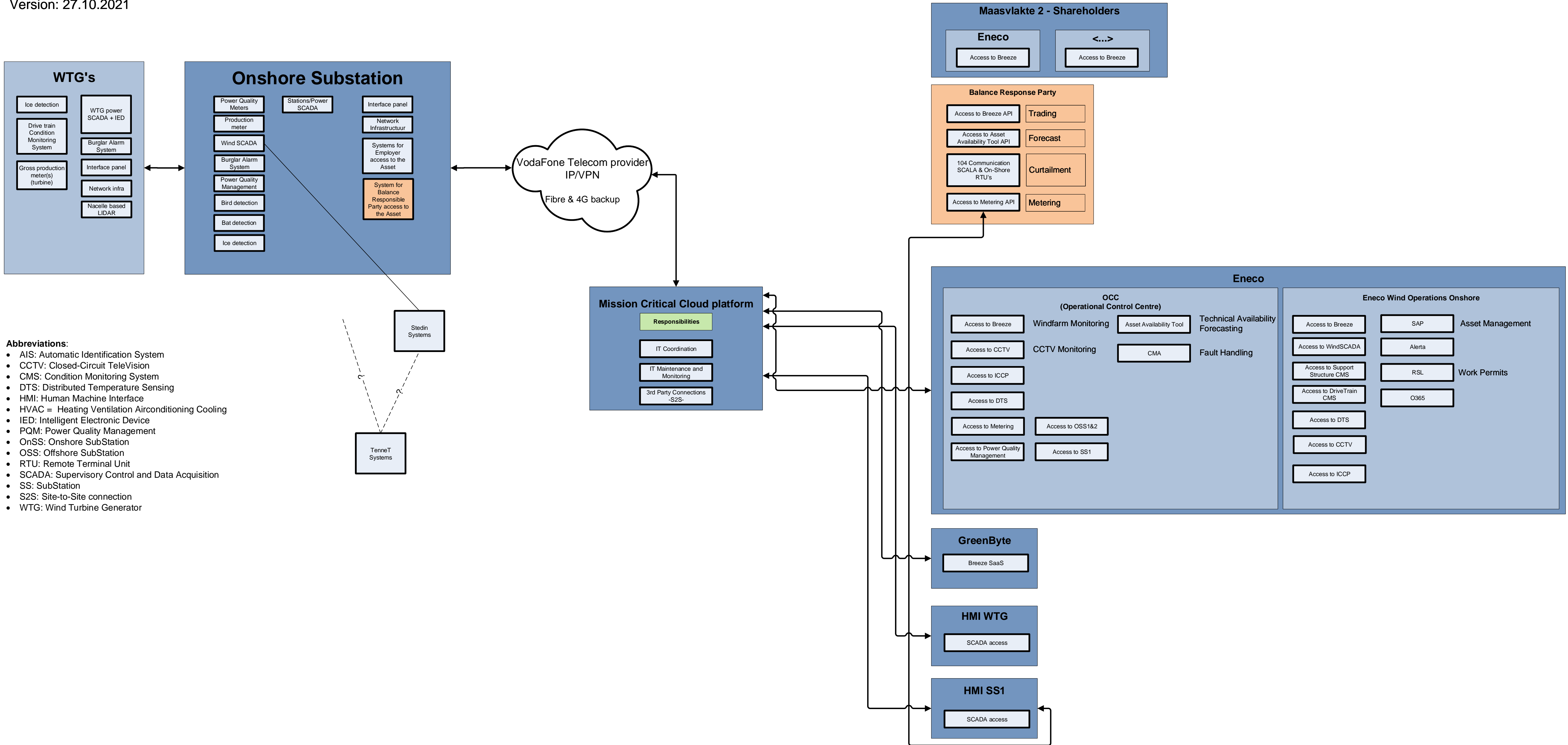
Om de communicatie van beschikbaarheid met de balanspartner mogelijk te maken vraagt Eneco van de balanspartner het volgende:

- (a) De balanspartner sluit aan op de Asset Availability Tool van Eneco, via de door Eneco beschikbaar gestelde Rest API (Annex 2).
- (b) De balanspartner gaat akkoord met het proces en de tijdslijnen voor communicatie van niet-beschikbaarheid zoals hierboven omschreven.

ANNEX 1: IT OVERZICHT

IT Overview Windfarm Maasvlakte II – the Netherlands

Version: 27.10.2021



- Abbreviations:**
- AIS: Automatic Identification System
 - CCTV: Closed-Circuit TeleVision
 - CMS: Condition Monitoring System
 - DTS: Distributed Temperature Sensing
 - HMI: Human Machine Interface
 - HVAC = Heating Ventilation Airconditioning Cooling
 - IED: Intelligent Electronic Device
 - PQM: Power Quality Management
 - OnSS: Onshore SubStation
 - OSS: Offshore SubStation
 - RTU: Remote Terminal Unit
 - SCADA: Supervisory Control and Data Acquisition
 - SS: SubStation
 - S2S: Site-to-Site connection
 - WTG: Wind Turbine Generator

ANNEX 2: WIND TURBINE AVAILABILITY API MANUAL V 2.3



windturbine availability forecasting API

VERSION

VITTERS, (HUGO), OOSTVEEN, MJ VAN (MAARTEN)



Content

Contacts.....	1
Version update.....	1
Audience	1
Information content	1
API General description	2
Request access token for external parties	2
Request access token for internal Eneco applications	3
Invoke windturbines-v1 API	3
Response message & Logic	4
SWAGGER 2.0.....	7

Contacts

Hugo Vitters	Expert operations Eneco	Hugo.vitters@eneco.com
Maarten van Oostveen	Product owner Integration team Eneco	Maarten.vanoostveen@eneco.com

Version update

2	27-Oct-21	Added functionality of restricted power output of powerplant (windfarm) by external influences
		Corrected description of logic behind "Limited" and "Interfered"
		Added extensive logic table

Audience

This document is aimed at external/ internal users of the Eneco wind availability service, which discloses the information of technical availability of windturbines via an API. This document contains the description of the API for integration purposes.

Information content

The API contains information about individual turbines and reflects there availability status, bases on 15 minute intervals and a 7-day look ahead horizon. The API also contains availability information about the power plant as a whole (windfarm)

- A user has only access to information about the windfarm/turbines agreed upon with Eneco
- A user has access to a ACC and Prod environment, only prod environment contains actual availability information
- A user receives login credentials through the Eneco Integration team.
- The API is a means of sharing information on a real-time basis
- The user is responsible for determining the frequency with which the API is invoked
- The 7 day look ahead does not necessarily conform with the maintenance horizon provided by the Asset management organization of the windfarm (usually a 5 days look ahead)



API General description

The windturbines-v1 API is secured with an access token (JWT) and apikey and accessible via a HTTPS-endpoint. In order to start using the API first a token needs to be obtained.

The API support two types of access tokens:

- Access tokens for external parties
- Access tokens for internal Eneco applications

The concept of requesting an access token is for both types the same, however there are technical differences.

Request access token for external parties

```
curl --request POST \  
  --url {{tokenendpoint}} \  
  --header 'accept: application/json' \  
  --header 'authorization: Basic MG9hY...' \  
  --header 'cache-control: no-cache' \  
  --header 'content-type: application/x-www-form-urlencoded' \  
  --data 'grant_type=client_credentials&scope=read'
```

Tokenendpoint:

- Acceptance environment:
<https://inloggen.acc.eneco.nl/oauth2/aus3enqfi9dfu7odf0i7/v1/token>
- Production environment:
<https://inloggen.eneco.nl/oauth2/aus3klmuv7x5CbV2b0i7/v1/token>

The Client ID and Secret aren't included in the POST body, but rather are placed in the HTTP Authorization header following the rules of HTTP Basic Auth.

Credentials will be different per environment.

Keep the HTTP Basic Auth credentials always secure!

Parameters that are being passed:

- `grant_type` is `client_credentials`, indicating that we are using the Client Credentials grant type.
- `scope` must be at least one scope. Use `read`.

If the credentials are valid the authorization server will return an access token:

```
{  
  "access_token": "eyJhbG[...]lLQ",  
  "token_type": "Bearer",  
  "expires_in": 3600,  
  "scope": "read"  
}
```

To obtain all details related to the authorization server please visit:

- acceptance environment:
<https://inloggen.acc.eneco.nl/oauth2/aus3enqfi9dfu7odf0i7/.well-known/oauth-authorization-server>
- Production environment:
<https://inloggen.eneco.nl/oauth2/aus3klmuv7x5CbV2b0i7/.well-known/oauth-authorization-server>



Request access token for internal Eneco applications (not applicable for Non-Eneco users)

```
POST https://login.microsoftonline.com/eneco.onmicrosoft.com/oauth2/token
Request header
  Content-Type:application/x-www-form-urlencoded
Request body
  grant_type:client_credentials
  client_id:{{client_id}}
  client_secret:{{secret}}
  resource:{{resource}}
```

Use `{{client_id}}` and `{{secret}}` to authenticate. The `{{resource}}` is depending on the environment:

- Acceptance environment:
api://f5f2e116-fe40-49f5-8f20-fc38d052fdf1
- Production environment:
api://5c685948-912f-4e2c-b1dd-ca91a688c527

Credentials will be different per environment.

Keep the credentials always secure!

If the credentials are valid the authorization server will return an access token:

```
{
  "token_type": "Bearer",
  "expires_in": "3599",
  "ext_expires_in": "3599",
  "expires_on": "1600425973",
  "not_before": "1600422073",
  "resource": "api://[...]",
  "access_token": "eyJ0eXAi[...]wNwycaQstgonw"
}
```

Invoke windturbines-v1 API

In combination with a valid apikey and access token you can start consuming the API.

- Be aware that the access token is only valid for a short period. The token can, during its validity period, be re-used. The API-endpoint will reject requests with an expired (or otherwise invalid) token. If you decide to re-use the access token in multiple API requests (which is a good idea for performance reasons) be aware that the lifetime (expiry period) of newly granted access tokens by the authorization server may change anytime.
- Usually the apikey won't expire (so can be re-used in all API requests).

Hostnames:

- Acceptance environment:
<https://api-acc.enecogroup.com/>
- Production environment:
<https://api.enecogroup.com/>

Full details about base-paths, supported operations and input/output structure can be found in the attached Swagger.



Response message & Logic

Table below contains names as supplied by the AAT. First in a logical model, then an example in JSON format Data Delivery (1)

Naam veld	Type	Description	meaning
	All timestamps in UTC		
dateTimeStart	String yyyy-mm-ddThh:mi:ssZ	Date starting period in quarters	Start of a specific quarter
dateTimeEnd	String yyyy-mm-ddThh:mi:ssZ	End date in quarters	End of a specific quarter
WindFarmExternalId	Integer	Windfarm Id	Id attributed to windfarm by Eneco
WindFarmName	String (50)	Windpark name	Commonly agreed windfarm name
TurbineSerialNummer	Integer	Turbine Id	Turbine ID as specified in mapping table
NotInterfered	Decimal	Capacity Turbine If 0 then attribute not present	Standard power capacity of the turbine (MW)
Limited	Decimal	Limited capacity Turbine If 0 then attribute not present	Amount of power available due to limitation of output or production (MW)
Interfered	Decimal	Standard capacity of Turbine when limited is active	Is only present when a turbine has limited power available as Interfered=notinterferd
assetRestrictedPowerOutput	Decimal	Available power output when windfarmt (asset) has restricted power output	Is only present when the windfarm has a restriction on power output, contains max available power both when determined by output restriction or by max. power of available turbines
CountryAbbreviation	String(2)	Afk. Land bv NL	



windturbine availability forecasting API, doc.version: 2

Logic table					
JSON payload logic	Notinterfered	Limited	Interfered	assetrestrictedpoweroutput	example

contains the standard capacity of turbine when fully available, not present when status of availability changes

contains the limitation on a turbine when he turbine has limited capacity due to malfunction or maintenance

contains the standard capacity of turbine when Limited is active

occurs when the powerplant as a whole has a restricted power output, caused by external influences, p.e. limitation due to overheating cables

Turbine status

Windturbine 100% available

present

absent

absent

Windturbine 100 % not available

absent

absent

absent

Windturbine partially available

absent

present

present

when a 2,5 MW turbine is available, NotInterfered contains 2,5. Limited and interfered are absent when a 2,5 MW turbine is unavailable due to maintenance or malfunction, Notinterfered, Limited and Interfered are absent. When a 2,5 MW turbine has a capacity limitation of 1 MW, Notinterfered is absent, Limited contains "1" and Interfered contains "2.5"

Powerplant has restricted power output



Total available turbine capacity >= PP output restriction	determined by turbine status	determined by turbine status	determined by turbine status	yes,contains restricted power output capacity	when Power Plant export capacity is limited at 450 MW and total available turbine capacity is 600MW, this field contains 450MW when Power Plant export capacity is limited at 450 MW, and total available turbine capacity =350, this field contains 350MW when Power Plant export capacity =0, this field contains "0.00000001"MW
Total available turbine capacity <= PP output restriction	determined by turbine status	determined by turbine status	determined by turbine status	yes, total of available turbine capacity	
PP remaining output =0				yes, contains "0.00000001"	



windturbine availability forecasting API, doc.version: 2

SWAGGER 2.0

```
{
  "swagger": "2.0",
  "info": {
    "version": "1.1.0",
    "title": "windturbines",
    "description": "",
    "contact": {
      "name": "Eneco Integration Team",
      "url": "https://developer.enecogroup.com",
      "email": "FM_BTO_Integration_Team@eneco.com"
    }
  },
  "host": "api.enecogroup.com",
  "basePath": "/",
  "schemes": [
    "https"
  ],
  "paths": {
    "/v1/windturbines/turbineforecast": {
      "get": {
        "tags": [
          "AvailableWindTurbineService"
        ],
        "description": "",
        "operationId": "turbineforecast",
        "produces": [
          "application/json"
        ],
        "parameters": [
          {
            "name": "x-request-id",
            "in": "header",
```



windturbine availability forecasting API, doc.version: 2

```
"type": "string",
"format": "uuid",
"description": "Unique identifier of the request for tracing purposes"
},
{
  "name": "datetimestart",
  "in": "query",
  "required": true,
  "type": "string",
  "format": "date-time",
  "x-example": "2014-12-31T23:59:59.938Z"
},
{
  "name": "datetimeend",
  "in": "query",
  "required": true,
  "type": "string",
  "format": "date-time",
  "x-example": "2014-12-31T23:59:59.938Z"
}
],
"responses": {
  "200": {
    "description": "",
    "schema": {
      "$ref": "#/definitions/APIForecastDataPerTurbine"
    }
  },
  "401": {
    "description": "Unauthorized",
    "schema": {
      "$ref": "#/definitions/errorResponseBody"
    }
  }
},
}
```



windturbine availability forecasting API, doc.version: 2

```
"403": {
  "description": "Forbidden",
  "schema": {
    "$ref": "#/definitions/errorResponseBody"
  }
},
"404": {
  "description": "Not Found",
  "schema": {
    "$ref": "#/definitions/errorResponseBody"
  }
},
"405": {
  "description": "Method Not Allowed",
  "schema": {
    "$ref": "#/definitions/errorResponseBody"
  }
},
"429": {
  "description": "Too Many Requests",
  "schema": {
    "$ref": "#/definitions/errorResponseBody"
  }
}
}
},
"definitions": {
  "APIForecastDataPerTurbine": {
    "description": "",
    "type": "object",
    "properties": {
      "data": {
```



windturbine availability forecasting API, doc.version: 2

```
"type": "array",
"items": {
  "$ref": "#/definitions/ForecastDataPerTurbine"
}
},
"errors": {
  "type": "array",
  "items": {
    "$ref": "#/definitions/error"
  }
}
},
"ForecastDataPerTurbine": {
  "description": "",
  "type": "object",
  "properties": {
    "dateTimeStart": {
      "type": "string",
      "format": "date-time",
      "example": "2014-12-31T23:59:59.938Z"
    },
    "dateTimeEnd": {
      "type": "string",
      "format": "date-time",
      "example": "2014-12-31T23:59:59.938Z"
    },
    "windFarmExternalId": {
      "type": "integer",
      "example": 0,
      "format": "int32"
    },
    "windFarmName": {
      "type": "string",
```



```
"default": ""
},
"turbineSerialNumber": {
  "type": "string",
  "default": ""
},
"notInterfered": {
  "type": "number",
  "example": 0.1
},
"limited": {
  "type": "number",
  "example": 0.1
},
"interfered": {
  "type": "number",
  "example": 0.1
},
"assetRestrictedPowerOutput": {
  "type": "number",
  "example": 0.1
},
"countryAbbreviation": {
  "type": "string",
  "default": ""
}
}
},
"error": {
  "description": "an individual error item, that can be listed",
  "type": "object",
  "properties": {
    "code": {
      "type": "string",
```



windturbine availability forecasting API, doc.version: 2

```
"default": "",
"description": "The errorcode"
},
"detail": {
  "type": "string",
  "default": "",
  "description": "the description of the error"
}
},
"errorResponseBody": {
  "description": "Response object containing errors",
  "type": "object",
  "properties": {
    "errors": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/error"
      }
    }
  }
},
"tags": [
  {
    "name": "AvailableWindTurbineService"
  }
],
"security": [
  {
    "apikey": [],
    "Okta": [],
    "Bearer": []
  }
]
```



windturbine availability forecasting API, doc.version: 2

```
],  
"securityDefinitions": {  
  "Okta": {  
    "name": "Authorization",  
    "in": "header",  
    "type": "apiKey",  
    "description": "Okta access token."  
  },  
  "apikey": {  
    "name": "apikey",  
    "in": "header",  
    "type": "apiKey",  
    "description": "Apigee API key"  
  },  
  "Bearer": {  
    "name": "Bearer",  
    "in": "header",  
    "type": "apiKey",  
    "description": "Azure Active Directory access token"  
  }  
}  
}
```