SGCWEBSOCKETS · TECHNICAL DOCUMENT

Pinecone Vector Database

Pinecone client — upsert, query and delete vectors in a managed Pinecone serverless or pod-based index from Delphi.

eSeGeCe

WWW.ESEGECE.COM

Overview

The component is based on the REST Pinecone API client which allows you to create / update / delete indexes and vectors.

At a glance

COMPONENT CLASS `TsgcAIDatabaseVectorPinecone`	STANDARDS / SPEC **Pinecone API reference**
TRANSPORTS **TCP, TLS**	PLATFORMS **Windows, macOS, Linux, iOS, Android**
FRAMEWORKS **VCL, FireMonkey, Lazarus / FPC**	EDITION **Standard / Professional / Enterprise**

Features

- Native Delphi implementation with full ANSI/Unicode support.

Technical specification

Standards & specs	Pinecone API reference · Pinecone quickstart
Component class	`TsgcAIDatabaseVectorPinecone` (unit `sgcAI_Database_Vector_Pinecone`)
Frameworks	VCL, FireMonkey, Lazarus / FPC
Platforms	Windows, macOS, Linux, iOS, Android

Main properties

The principal published / public properties used to configure and drive the component. Consult the online help for the full list.

`PineconeOptions`	Pinecone account credentials and HTTP client options shared by all requests.
`PineconeIndexOptions`	Target Pinecone index and project used by upsert and query operations.
`Version`	Read-only sgcWebSockets library version string.

Main methods

The principal public methods exposed by the component.

`QueryData()`	Runs a similarity query against the Pinecone index and returns the raw JSON response.
`BeginAddData()`	Opens a batch of vectors that will be sent to Pinecone in a single upsert call.
`AddData()`	Queues a single vector inside the current Pinecone upsert batch.
`EndAddData()`	Sends the accumulated vectors to Pinecone in a single upsert request.

Quick Start

Drop the component on a form, configure the properties below and activate it. The snippet that follows shows the typical **Example UPSERT** configuration sourced from the online help.

> **About this scenario.** Find below an example of UPSERT a single vector with the Id = "id1".

Delphi (VCL / FireMonkey)

```
procedure UpsertPinecone(const aIndexName, aProjectId: string; const aVector: Array of Double);
var
  oPinecone: TsgcHTTP_API_Pinecone;
  oParams: TsgcHTTPPineconeVectorUpserts;
  oVectors: TsgcArrayOfVectorUpsert;
begin
  oPinecone := TsgcHTTP_API_Pinecone.Create(nil);
  Try
    oPinecone.PineconeOptions.API := 'your-api-key';
    oParams := TsgcHTTPPineconeVectorUpserts.Create;
    Try
      SetLength(oVectors, 1);
      oVectors[0] := TsgcHTTPPineconeVectorUpsert.Create;
      oVectors[0].Id := 'id1';
      oVectors[0].Values := aVector;
      oParams.Vectors := oVectors;
      Pinecone.VectorsUpsert(aIndexName, aProjectId, oParams);
    Finally
      oParams.Free;
    End;
  Finally
    oPinecone.Free;
  End;
end;
```

C++ Builder

```
void UpsertPinecone(const String aIndexName, const String aProjectId, const std::vector<double> &
{
  TsgcHTTP_API_Pinecone* oPinecone = new TsgcHTTP_API_Pinecone(NULL);
  try
  {
    oPinecone->PineconeOptions.API = "your-api-key";
    TsgcHTTPPineconeVectorUpserts* oParams = new TsgcHTTPPineconeVectorUpserts();
    try
    {
      TsgcArrayOfVectorUpsert oVectors;
      oVectors.push_back(new TsgcHTTPPineconeVectorUpsert());
      oVectors[0]->Id = "id1";
      oVectors[0]->Values = aVector;
      oParams->Vectors = oVectors;
      oPinecone->VectorsUpsert(aIndexName, aProjectId, oParams);
    }
    __finally
    {
      oParams->Free();
    }
  }
  __finally
  {
    oPinecone->Free();
  }
}
```

.NET (C#)

```
void UpsertPinecone(string aIndexName, string aProjectId, double[] aVector)
{
    TsgcHTTP_API_Pinecone oPinecone = new TsgcHTTP_API_Pinecone(null);
    oPinecone.PineconeOptions.API = "your-api-key";
    TsgcHTTPPineconeVectorUpserts oParams = new TsgcHTTPPineconeVectorUpserts();
    TsgcArrayOfVectorUpsert oVectors = new TsgcArrayOfVectorUpsert();
    oVectors.Add(new TsgcHTTPPineconeVectorUpsert()
    {
        Id = "id1",
        Values = aVector
    });
    oParams.Vectors = oVectors;
    oPinecone.VectorsUpsert(aIndexName, aProjectId, oParams);
}
```

Common scenarios

The following scenarios are lifted verbatim from the online help. Each shows the configuration and method calls needed to drive the component through a specific real-world flow.

1 · Example QUERY

Find below an example of QUERY a single vector.

Delphi (VCL / FireMonkey)

```
procedure QueryPinecone(const aIndexName, aProjectId: string; const aVector: Array of Double);
var
  oParams: TsgcHTTPPineconeVectorQuery;
begin
  oParams := TsgcHTTPPineconeVectorQuery.Create;
  Try
    oParams.Vector := aVector;
    Pinecone.VectorsQuery(aIndexName, aProjectId, oParams);
  Finally
    oParams.Free;
  End;
end;
```

C++ Builder

```
void QueryPinecone(const string aIndexName, const string aProjectId, const std::vector<double>& 
{
  TsgcHTTPPineconeVectorQuery* oParams = new TsgcHTTPPineconeVectorQuery();
  try
  {
    oParams->Vector = aVector;
    Pinecone.VectorsQuery(aIndexName, aProjectId, oParams);
  }
  __finally
  {
    oParams->Free();
  }
}
```

.NET (C#)

```
void QueryPinecone(string aIndexName, string aProjectId, double[] aVector)
{
    TsgcHTTPPineconeVectorQuery oParams = new TsgcHTTPPineconeVectorQuery();
    oParams.Vector = aVector;

    Pinecone.VectorsQuery(aIndexName, aProjectId, oParams);
}
```

Sources used to build this document

Every external claim links back to a primary source. The online-help references decode the canonical deep-link the company maintains for this component.

Primary standard / spec — Pinecone API reference	docs.pinecone.io/reference/api/introduction
Primary standard / spec — Pinecone quickstart	docs.pinecone.io/guides/get-started/quickstart
Online help — component page	www.esegece.com/help/sgcWebSockets/Components/AI/Applications/Embeddings/Databases/TsgcAIDatabaseVectorPinecone.htm
Delphi demo project (in the sgcWebSockets package)	`Demos\15.AI\10.Vector_Database\01.Pinecone`
Component page	www.esegece.com/products/websockets/ai/pinecone/
Product page	www.esegece.com/products/websockets/

> **Document scope.** This document covers the publicly-documented surface of the Pinecone Vector Database component shipped with sgcWebSockets. For full property, method and event reference consult the online help linked above.