```
library TaskAPI;

uses
  Winapi.ActiveX,
  System.Win.ComObj,
  Web.WebBroker,
  Web.Win.ISAPIApp,
  Web.Win.ISAPIThreadPool,
  WebModuleUnit in 'WebModuleUnit.pas' {WebModule1: TWebModule};

{$R *.res}

exports
  GetExtensionVersion,
  HttpExtensionProc,
  TerminateExtension;

begin
  CoInitFlags := COINIT_MULTITHREADED;
  Application.Initialize;
  Application.WebModuleClass := WebModuleClass;
  Application.Run;
end.

procedure TWebModule1.WebModuleCreate(Sender: TObject);
begin
  FDConnection1.DriverName := 'PG';
  FDConnection1.Params.Values['Server'] := 'localhost';
  FDConnection1.Params.Values['Database'] := 'task_manager';
  FDConnection1.Params.Values['User_Name'] := 'postgres';
  FDConnection1.Params.Values['Password'] := 'ваш_пароль';
  FDConnection1.Params.Values['CharacterSet'] := 'UTF8'; // Для поддержки кириллицы
  FDConnection1.Connected := True; // Активируйте подключение
end;

procedure TWebModule1.WebModule1TasksGetAction(Sender: TObject;
  Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);
var
  Query: TFDQuery;
  JSONArray: TJSONArray;
begin
  Query := TFDQuery.Create(nil);
  try
    Query.Connection := FDConnection1;
    Query.SQL.Text := 'SELECT * FROM tasks ORDER BY task_id';
    Query.Open;

    JSONArray := TJSONArray.Create;
    try
      while not Query.Eof do
      begin
        JSONArray.AddElement(
          TJSONObject.Create
            .AddPair('id', Query.FieldByName('task_id').AsInteger)
            .AddPair('title', Query.FieldByName('title').AsString)
            .AddPair('description', Query.FieldByName('description').AsString)
            .AddPair('status', Query.FieldByName('status').AsString)
            .AddPair('due_date', FormatDateTime('yyyy-mm-dd',
Query.FieldByName('due_date').AsDateTime))
        );
        Query.Next;
      end;
      Response.Content := JSONArray.ToString;
    finally
      JSONArray.Free;
    end;
  finally
    Query.Free;
  end;
  Response.ContentType := 'application/json';
  Handled := True;
end;

procedure TWebModule1.WebModule1TasksPostAction(Sender: TObject;
  Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);
var
  JSON: TJSONObject;
  Query: TFDQuery;
begin
  JSON := TJSONObject.ParseJSONValue(Request.Content) as TJSONObject;
  try
    Query := TFDQuery.Create(nil);
    try
      Query.Connection := FDConnection1;
      Query.SQL.Text :=
        'INSERT INTO tasks (title, description, status, due_date) ' +
        'VALUES (:title, :desc, :status, :due_date) RETURNING task_id';

      Query.ParamByName('title').Value := JSON.GetValue('title').Value;
      Query.ParamByName('desc').Value := JSON.GetValue('description').Value;
      Query.ParamByName('status').Value := 'Pending'; // Статус по умолчанию
      Query.ParamByName('due_date').Value :=
        StrToDate(JSON.GetValue('due_date').Value); // Формат: 'YYYY-MM-DD'

      Query.Open;
      Response.Content :=
        TJSONObject.Create
          .AddPair('id', Query.FieldByName('task_id').AsInteger)
          .AddPair('status', 'created')
          .ToString;
    finally
      Query.Free;
    end;
  finally
    JSON.Free;
  end;
  Response.ContentType := 'application/json';
  Response.StatusCode := 201; // 201 Created
  Handled := True;
end;

                procedure TWebModule1.WebModule1TaskPutAction(Sender: TObject;
  Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);
var
  TaskID: Integer;
  JSON: TJSONObject;
  Query: TFDQuery;
begin
  TaskID := StrToInt(Request.PathInfo.Substring(7)); // Извлекаем ID из /tasks/123
  JSON := TJSONObject.ParseJSONValue(Request.Content) as TJSONObject;

  try
    Query := TFDQuery.Create(nil);
    try
      Query.Connection := FDConnection1;
      Query.SQL.Text :=
        'UPDATE tasks SET ' +
        'title = :title, ' +
        'description = :desc, ' +
        'status = :status, ' +
        'due_date = :due_date ' +
        'WHERE task_id = :id';

      Query.ParamByName('id').Value := TaskID;
      Query.ParamByName('title').Value := JSON.GetValue('title').Value;
      Query.ParamByName('desc').Value := JSON.GetValue('description').Value;
      Query.ParamByName('status').Value := JSON.GetValue('status').Value;
      Query.ParamByName('due_date').Value :=
        StrToDate(JSON.GetValue('due_date').Value);

      Query.ExecSQL;

      Response.Content :=
      TJSONObject.Create
        .AddPair('status', 'updated')
        .ToString;
    finally
      Query.Free;
    end;
  finally
    JSON.Free;
  end;
  Response.ContentType := 'application/json';
  Handled := True;
end;

procedure TWebModule1.WebModule1TaskDeleteAction(Sender: TObject;
  Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);
var
  TaskID: Integer;
  Query: TFDQuery;
begin
  TaskID := StrToInt(Request.PathInfo.Substring(7)); // Извлекаем ID из /tasks/123

  Query := TFDQuery.Create(nil);
  try
    Query.Connection := FDConnection1;
    Query.SQL.Text := 'DELETE FROM tasks WHERE task_id = :id';
    Query.ParamByName('id').Value := TaskID;
    Query.ExecSQL;

    Response.Content :=
      TJSONObject.Create
        .AddPair('status', 'deleted')
        .ToString;
  finally
    Query.Free;
  end;
  Response.ContentType := 'application/json';
  Handled := True;
end;

procedure TWebModule1.WebModule1Exception(Sender: TObject;
  Request: TWebRequest; Response: TWebResponse; var Handled: Boolean);
begin
  Response.Content :=
    TJSONObject.Create
      .AddPair('error', Exception(ExceptObject).Message)
      .ToString;
  Response.ContentType := 'application/json';
  Response.StatusCode := 500;
  Handled := True;
end;
```